

Introduction aux modèles mixtes

P^r Jean R. LOBRY

On part d'un jeu de données bruitées dont on maîtrise entièrement les déterminants : l'évolution au cours du temps, au voisinage de l'équinoxe du printemps, de l'heure de coucher du soleil dans divers sites en EUROPE. Après avoir exploré ce jeu de données pour l'avoir bien en main, on dégrade l'information en faisant comme si la géolocalisation de nos sites d'observation était inconnue. On analyse alors ce jeu de données dégradées avec des modèles mixtes en introduisant les sites en effet aléatoire tant au niveau de la pente que de l'ordonnée à l'origine.

Table des matières

1	Modèles linéaires classiques	3
1.1	La problématique	3
1.2	Voyage au centre du monde	3
1.3	Localisation des sites	7
1.4	Modèle linéaire local	9
1.5	Analyse des « données »	12
1.5.1	Effet de la longitude	12
1.5.2	Effet de la latitude	14
1.6	Modèle linéaire à réponse non linéaire	16
2	Le jeu de données dégradées	22
2.1	Dégradation totale de l'information	22
2.2	Dégradation partielle de l'information	22
2.3	Modèle linéaire global	23
3	Modèles linéaires mixtes	25
3.1	Syntaxe	25
3.2	Ordonnées en effet aléatoire	25
3.3	Pentes en effet aléatoire	29
3.4	Pentes et ordonnées en effet aléatoire	31
3.4.1	Pentes et ordonnées corrélées	32
3.4.2	Pentes et ordonnées non corrélées	34
	Références	39

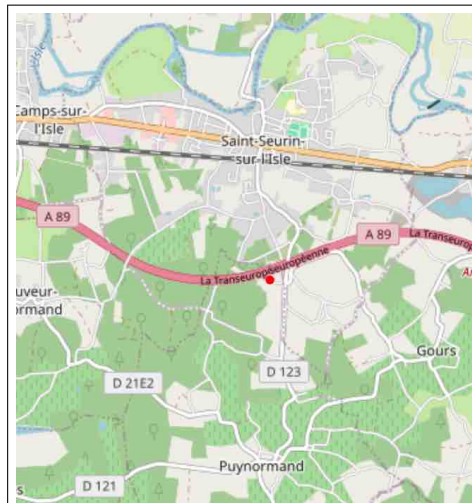


FIGURE 1 : Le point rouge est au centre du monde à la longitude 0° et à la latitude 45° .

1 Modèles linéaires classiques

1.1 La problématique

LA question que nous posons ici est simple : au printemps, de combien augmente chaque jour, en moyenne, l'heure de coucher du soleil ? Autrement dit, on aimerait un modèle qui nous prédise, en fonction de la date, l'heure de la tombée de la nuit. Le paquet `R suncalc` [15] nous permet de répondre de façon *exacte* à cette question, nous allons l'utiliser de façon *dégradée* pour simuler des campagnes d'observation.

UNE expédition coûte cher, nous avons besoin de faire une expérience préliminaire pour convaincre nos bailleurs de fonds. Pour être le plus représentatif possible, nous allons nous rendre au centre du monde : à la longitude 0 de nos GPS et à mi-distance angulaire entre le pôle et l'équateur.

1.2 Voyage au centre du monde

POUR voyager virtuellement au centre du monde nous utilisons le paquet `terra` [8] pour produire une carte interactive avec le code ci-après. Nous sommes donc au nord de la commune de PUYNORMAND, au sud de SAINT-SEURIN SUR L'ISLE, dans le département de la GIRONDE (figure 1 page 3). On trouvera sur le site¹ du *degree confluence projet* de nombreuses photos de cet endroit si particulier (*e.g.* figure 2 page 4).

```
library(terra)
cdm <- vect(cbind(0.0, 45.0), crs = "EPSG:4326")
plot(cdm, col = "red")
```

¹<https://confluence.org>



FIGURE 2 : Photo prise le 12 juin 2025 au centre du monde postée par H. PIERRICK sur le site du *degre conflunce projet*. Le poteau indique le méridien de GREENWICH, mais je suspecte qu'il s'agit en fait du méridien de référence de nos GPS, et non du méridien de GREENWICH *sensu stricto* qui se trouve à 5.3 secondes d'arc à l'ouest du méridien de référence, soit de l'ordre d'une centaine de mètres de décalage à cette latitude.

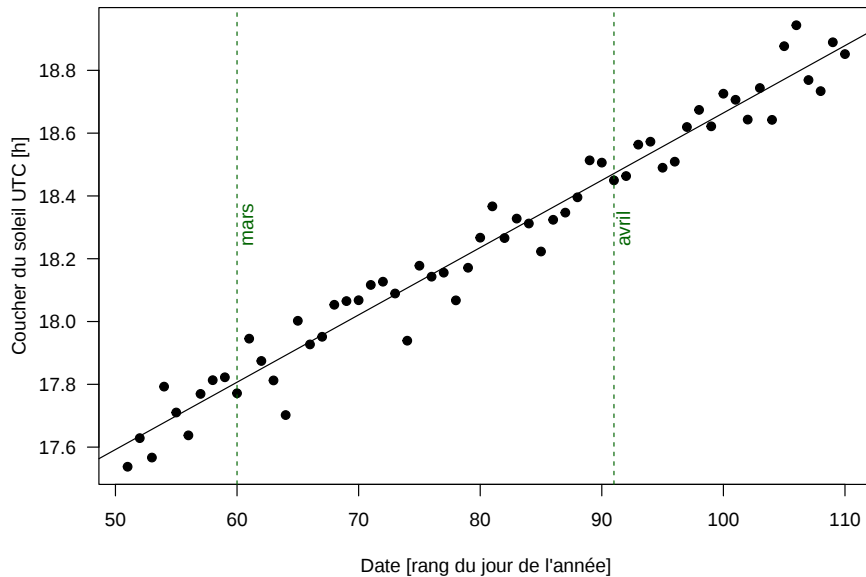
EN 2026 l'équinoxe de printemps tombait le 20 mars, nous décidons donc de faire une campagne d'observation du 20 février au 20 avril (soit 60 jours d'observation). Pour être plus réaliste, nous dégradons le calcul exact en ajoutant un bruit gaussien d'écart-type de 5 minutes. C'est la fonction `cds()` qui se charge des calculs.

```
library(suncalc)
cgs <- function(date, lat, lon, sd = 5/60){
  require(lubridate)
  n <- length(date)
  dsr <- getSunlightTimes(date, lat, lon, keep = c("sunset"))
  res <- with(dsr, hour(sunset) + minute(sunset)/60 + second(sunset)/3600 + rnorm(n, sd = sd))
  return(res)
}
cgs(as.Date("2026-03-20"), 45, 0, sd = 0) # environ à 18 h zulu time
[1] 18.21111
```

NOUS utilisons maintenant cette fonction pour représenter l'évolution de l'heure de coucher du soleil au centre du monde. Pour simplifier, on représente la date par le rang du jour dans l'année calculée avec la fonction `ydate()` du paquet `lubridate` [7]. On ajoute des lignes pointillées verticales pour représenter le début du mois de mars et d'avril pour avoir des repères plus familiers. Les heures de coucher du soleil seront toujours ici dans le fuseau horaire UTC (le *zulu time* des militaires ou encore GMT).

```
tobs <- seq(as.Date("2026-02-20"), as.Date("2026-04-20"), by = 1) ; nt <- length(tobs)
set.seed(1) # pour reproductibilité
y <- cgs(tobs, 45.0, 0.0)
library(lubridate)
x <- yday(tobs)
plot(x, y, xlab = "Date [rang du jour de l'année]", las = 1,
     ylab = "Coucher du soleil UTC [h]", pch = 19,
     main = "Coucher du soleil à Puynormand au voisinage du printemps")
abline(lm1 <- lm(y~x))
adddebmois <- function(){
  debmois <- yday(as.Date(paste0("2026-0", 3:4, "-01")))
  myc <- "darkgreen"
  abline(v = debmois, lty = 2, col = myc)
  pu <- par("usr")
  text(debmois, (pu[3] + pu[4])/2, c("mars", "avril"), srt = 90, pos = 4, col = myc)
}
adddebmois()
ycdmrange <- range(y, na.rm = TRUE)
```

Coucher du soleil à Puynormand au voisinage du printemps



La tendance générale est à l'évidence très bien résumée par une droite où la variable quantitative dépendante Y est une fonction linéaire de la variable quantitative prédictive X plus du bruit ϵ :

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (1)$$

Le résumé du modèle est :

```
summary(lm1)
Call:
lm(formula = y ~ x)
Residuals:
    Min       1Q   Median       3Q      Max
-0.190353 -0.040837 -0.000768  0.049400  0.150593

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.652e+01  4.416e-02  374.12  <2e-16 ***
x             2.144e-02  5.363e-04   39.98  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07194 on 58 degrees of freedom
Multiple R-squared:  0.965,    Adjusted R-squared:  0.9644
F-statistic: 1599 on 1 and 58 DF,  p-value: < 2.2e-16
```

La pente de la droite vaut $\beta_1 \approx 0.021 \text{ h.j}^{-1}$ soit environ 1.3 minutes plus tard chaque jour pour l'heure de coucher du soleil. Le deuxième paramètre, l'ordonnée de la droite à l'origine des temps, vaut $\beta_0 \approx 16.5 \text{ h}$. Comme le temps est représenté ici par le rang du jour dans l'année, le $t = 0$ correspond en fait au 31 décembre de l'année précédente. L'extrapolation faite ici n'est pas complètement délirante :

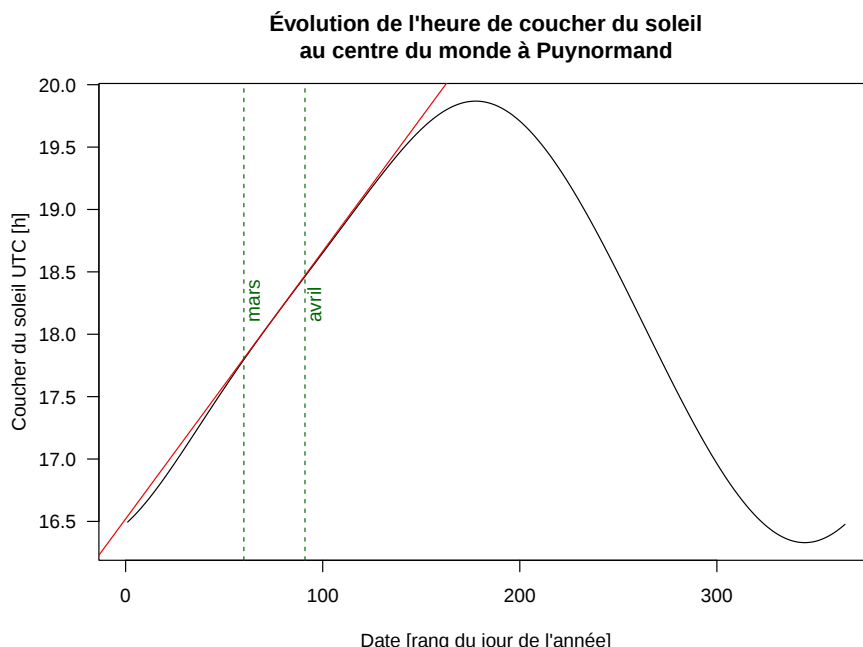
```
cds(as.Date("2025-12-31"), 45.0, 0.0, sd = 0)
[1] 16.48028
```

MAIS si on extrapole au 31 décembre de l'année en cours cela ne fait plus aucun sens avec une heure de coucher du soleil supérieure à 24 heures :

```
predict(lm1, list(x = yday(as.Date("2026-12-31"))))
1
24.34669
```

IL est quand même surprenant d'avoir un modèle linéaire aussi bon ($r^2 = 0.97$) alors que l'on s'attendrait à avoir une réponse plus ou moins sinusoïdale. Pour comprendre ce qu'il se passe on représente l'évolution de l'heure de coucher du soleil au cours de toute l'année.

```
allYear <- seq(as.Date("2026-01-01"), as.Date("2026-12-31"), by = 1)
cds2026 <- cds(allYear, 45, 0, sd = 0)
plot(yday(allYear), cds2026, type = "l", las = 1,
     main = "Évolution de l'heure de coucher du soleil\neau centre du monde à Puynormand",
     xlab = "Date [rang du jour de l'année]",
     ylab = "Coucher du soleil UTC [h]")
abline(lm1, col = "red2")
adddebmoids()
```



NOTRE modèle linéaire en rouge est une très bonne approximation de la réalité au voisinage de l'équinoxe du printemps. L'équinoxe correspond au point d'inflexion de la courbe, là où l'approximation linéaire est la meilleure (on aurait la même chose à l'équinoxe d'automne). C'est un pour cela que l'on a décidé de travailler au printemps.

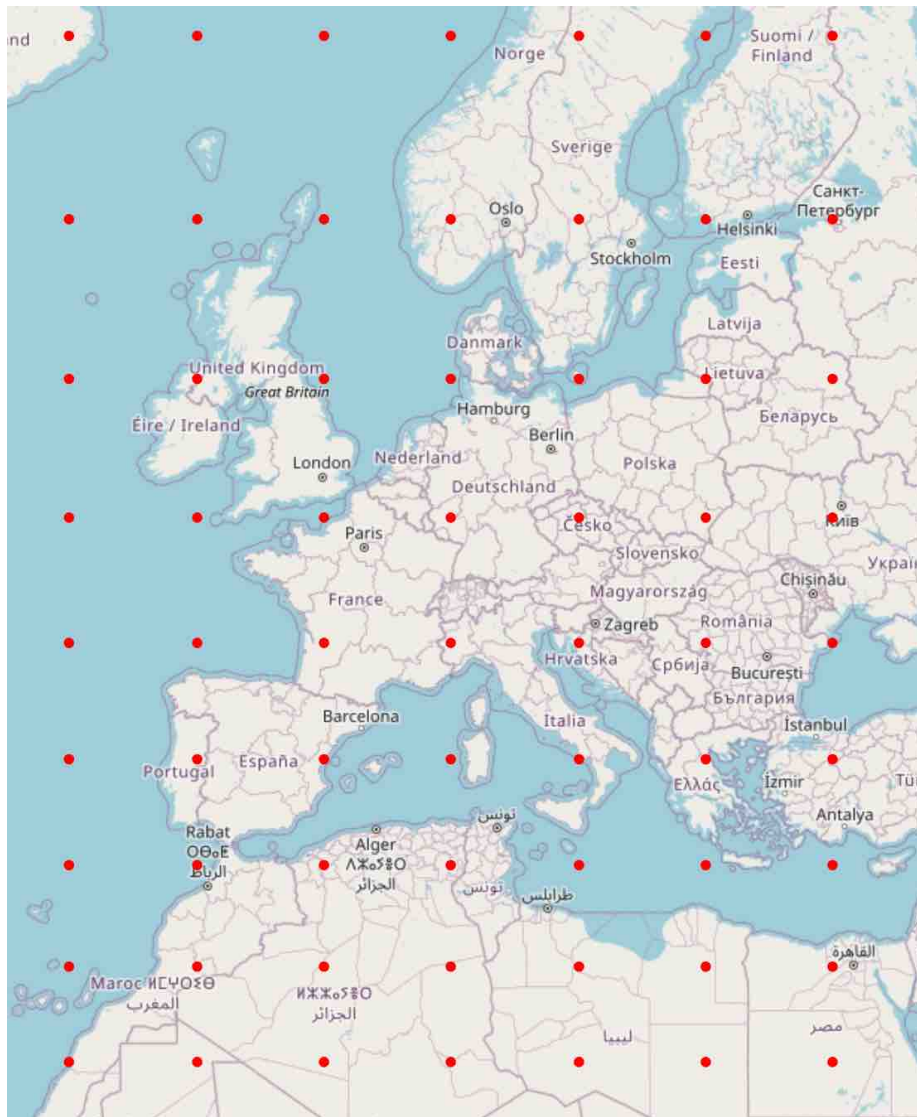
FORT de notre modèle, nous avons maintenant des arguments à faire valoir au près de nos investisseurs pour les convaincre de financer une campagne plus ambitieuse pour augmenter le nombre de sites suivis.

1.3 Localisation des sites

NOUS décidons de visiter 63 sites placés sur une grille régulière selon le modèle suivant : 9 latitudes en partant de 25° avec un pas de 5° et 7 longitudes en

partant de -15° avec un pas de 7.5° , soit un demi-fuseau horaire :

```
latseq <- seq(25, 65, by = 5) ; nlat <- length(latseq)
lonseq <- seq(-15, 35, by = 15/2) ; nlon <- length(lonseq)
pts <- vect(cbind(rep(lonseq, nlat), rep(latseq, each = nlon)), crs = "EPSG:4326")
plet(pts, col = "red")
```



NOUS faisons maintenant rouler notre fonction `cds()` sur tous les sites, et on en profite pour calculer au passage les coefficients de la droite de régression (eq. 1) au sein de chaque site, ce que nous appellerons par la suite le modèle linéaire local (section 1.4 page 9).

```
set.seed(1)
x <- xlon <- ylat <- y <- yt <- numeric(0)
cnames <- c("xlon", "ylat", "slp", "inter", "r2", "rms")
dfp <- as.data.frame(matrix(NA, nrow = nlon*nlat, ncol = length(cnames)))
colnames(dfp) <- cnames
ii <- 1
```

```
for(j in seq_len(nlon)){
  for(i in seq_len(nlat)){
    dfp[ii, "ylat"] <- latseq[i]
    dfp[ii, "xlon"] <- lonseq[j]
    yp <- cds(tobs, latseq[i], lonseq[j])
    xp <- yday(tobs)
    lm0 <- lm(yp~xp)
    dfp[ii, "slp"] <- 60.0*lm0$coef[2] # en min
    dfp[ii, "inter"] <- lm0$coef[1] # en h
    dfp[ii, "r2"] <- cor(yp, lm0$fitted.values)^2
    dfp[ii, "rms"] <- sqrt(mean(lm0$residuals^2))
    ii <- ii + 1
    x <- c(x, xp)
    ylat <- c(ylat, rep(latseq[i], nt))
    xlon <- c(xlon, rep(lonseq[j], nt))
    y <- c(y, yp)
    yt <- c(yt, lm0$fitted.values)
  }
}
df <- data.frame(list(y = y, x = x, xlon = xlon, ylat = ylat))
```

1.4 Modèle linéaire local

ON note n le nombre de sites, dans notre cas on a $n = 63$. Par modèle linéaire local on entend ici le modèle qui consiste à faire une droite de régression site par site, il comporte donc $2 \times 63 = 126$ paramètres. Pour un site s on note sX le vecteur de la variable de contrôle, le rang du jour dans l'année, et sY le vecteur des valeurs observées, soit les heures de coucher du soleil. On a donc une collection de n modèles linéaires, avec pour chaque site s son ordonnée à l'origine ${}^s\beta_0$ et sa pente de la droite de régression ${}^s\beta_1$:

$$\forall s \in \{1, 2, \dots, n\} : {}^sY = {}^s\beta_0 + {}^s\beta_1 {}^sX + \epsilon \quad (2)$$

LES ajustements sont fait indépendamment site par site, il n'y a aucune restriction sur les coefficients des droites de régression :

$${}^s\beta_0 \in \mathbb{R}^n \quad ; \quad {}^s\beta_1 \in \mathbb{R}^n \quad (3)$$

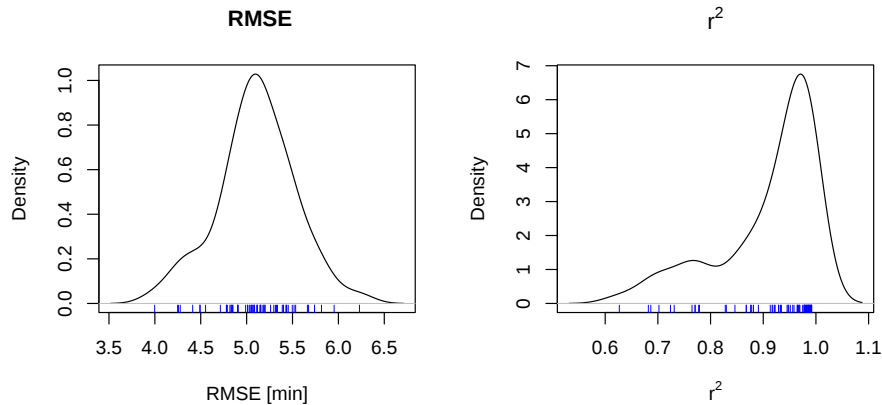
DANS les modèles mixtes, les paramètres ne seront plus libres de gambader dans $\mathbb{R}^n$ parce que les ajustements ne seront plus indépendants les uns des autres. Pour ne pas alourdir les notations, on va se contenter de colorier les paramètres en rouge pour signifier qu'il y a des contraintes à respecter. Voici par exemple la notation pour un modèle mixte dans lequel l'ordonnée à l'origine est mise en effet aléatoire (elle peut varier d'un site à l'autre) et la pente est mise en effet fixe (elle est commune à tous les sites) :

$$\forall s \in \{1, 2, \dots, n\} : {}^sY = {}^s\beta_0 + \beta_1 {}^sX + \epsilon \quad (4)$$

LA distribution des RMSE² du modèle linéaire local est centrée autour de 5 min, ce qui correspond à l'écart-type du bruit que nous avons introduit, c'est assez logique. La distribution des r^2 montre qu'ils sont souvent (42/63 = 67 %) supérieurs à 0.9, mais il reste un tiers des sites où ils ne le sont pas.

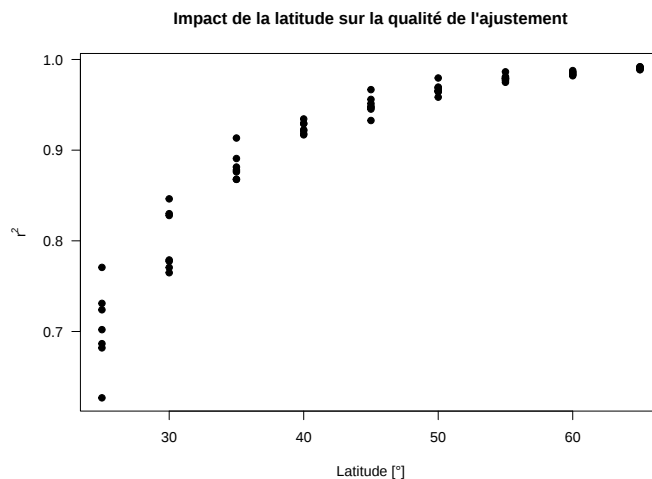
²On expliquera plus en détail ce que c'est dans la section 3.2 page 25.

```
mydst <- function(x, ...){ plot(dst <- density(x), ...) ; rug(x, col = "blue")}
par(mfrow = c(1, 2))
mydst(60*dfp$rms, main = " RMSE", xlab = "RMSE [min]")
mydst(dfp$r2, main = bquote(r^2), xlab = bquote(r^2))
```



POUR comprendre pourquoi, on représente la valeur des r^2 en fonction de la latitude. Il appert qu'il y a un impact majeur de cette dernière avec des r^2 plus élevés aux hautes latitudes :

```
plot(dfp$ylat, dfp$r2, ylab = bquote(r^2), xlab = "Latitude [°]", las = 1, pch = 19,
     main = "Impact de la latitude sur la qualité de l'ajustement")
```



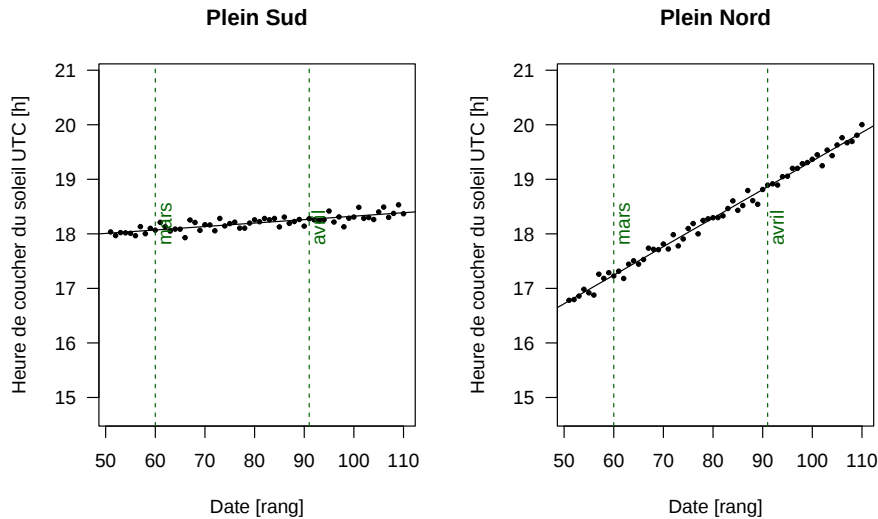
NOUS avons ici simplement une variation du rapport signal/bruit : l'amplitude de variation de l'heure de coucher du soleil est plus forte aux hautes latitudes. On peut visualiser ceci avec les deux sites de latitude extrême sur le méridien d'origine :

```
par(mfrow = c(1, 2))
plotxlonlat <- function(lon, lat, ...){
  with(subset(df, xlon == lon & ylat == lat), {
    plot(x, y, pch = 19, las = 1, cex = 0.5, xlab = "Date [rang]",
         ylab = "Heure de coucher du soleil UTC [h]", ...)
    abline(lm(y~x))
  })
}
```

```

    adddebmois()
  })
}
plotxlonylat(0, min(latseq), ylim = range(df$y), main = "Plein Sud")
plotxlonylat(0, max(latseq), ylim = range(df$y), main = "Plein Nord")

```



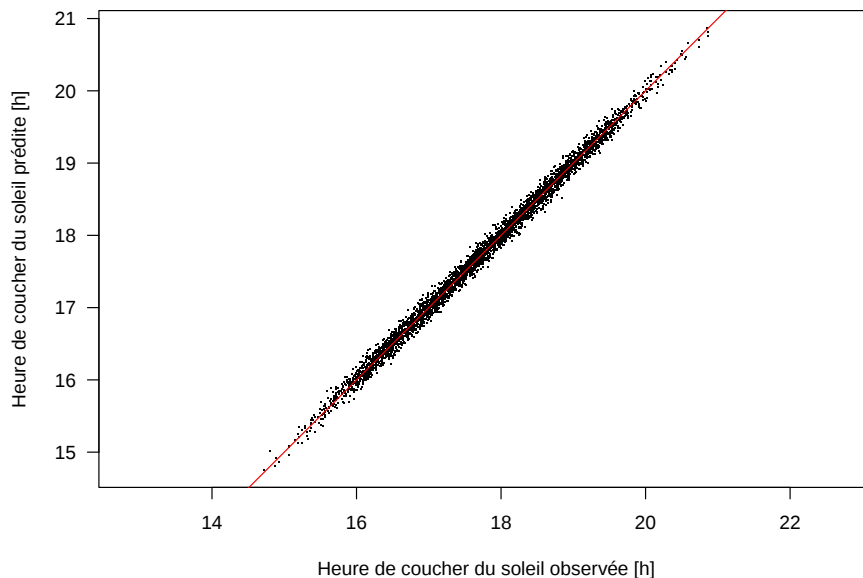
Le modèle linéaire local a deux intérêts. Premièrement, il nous donne la limite ultime de ce que l'on peut espérer en terme de qualité de l'ajustement aux données :

```

plot(y, yt, asp = 1, las = 1, main = "Le modèle linéaire local", pch = ".",
     xlab = "Heure de coucher du soleil observée [h]",
     ylab = "Heure de coucher du soleil prédite [h]")
abline(c(0, 1), col = "red2")

```

Le modèle linéaire local



MAIS sa limite est son incapacité à généraliser puisque que nous n'avons pas tenté d'utiliser les deux variables de contrôle que sont la latitude et la longitude. Secondement, il nous permet de résumer nos 3780 observations en un jeu de 126 paramètres, ce qui va nous faciliter l'analyse des « données. »

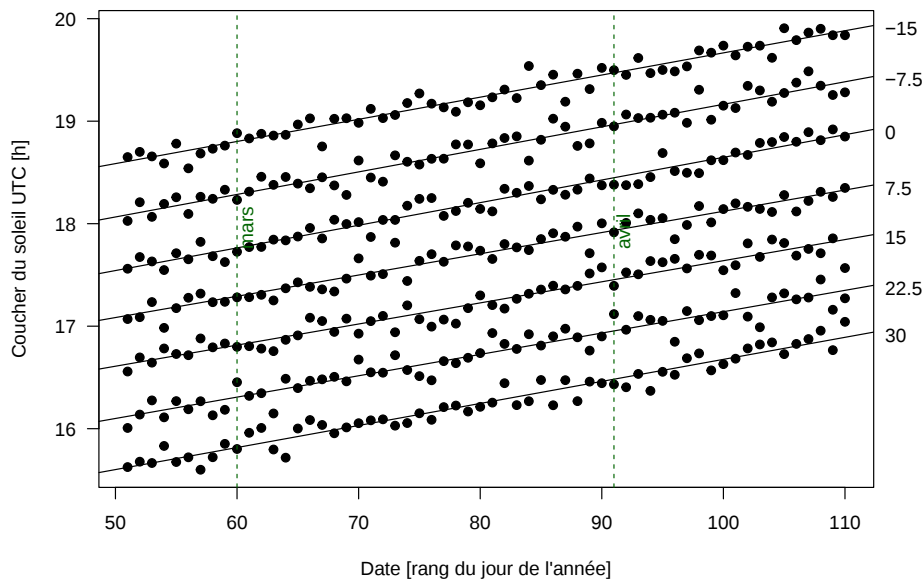
1.5 Analyse des « données »

1.5.1 Effet de la longitude

POUR mieux comprendre ce qu'il se passe on décide de regarder une série de site de même latitude. On voit que la réponse est simplement translatée en ordonnée. C'est ici simplement l'effet du décalage horaire : le soleil se levant à l'est, il va se coucher plus tard sur les sites les plus à l'ouest, aux faibles longitudes. Nous avons placés régulièrement nos site tous les 7.5° de longitude, soit un demi-fuseau horaire, il y a un décalage de 30 min entre deux sites voisins sur une même latitude.

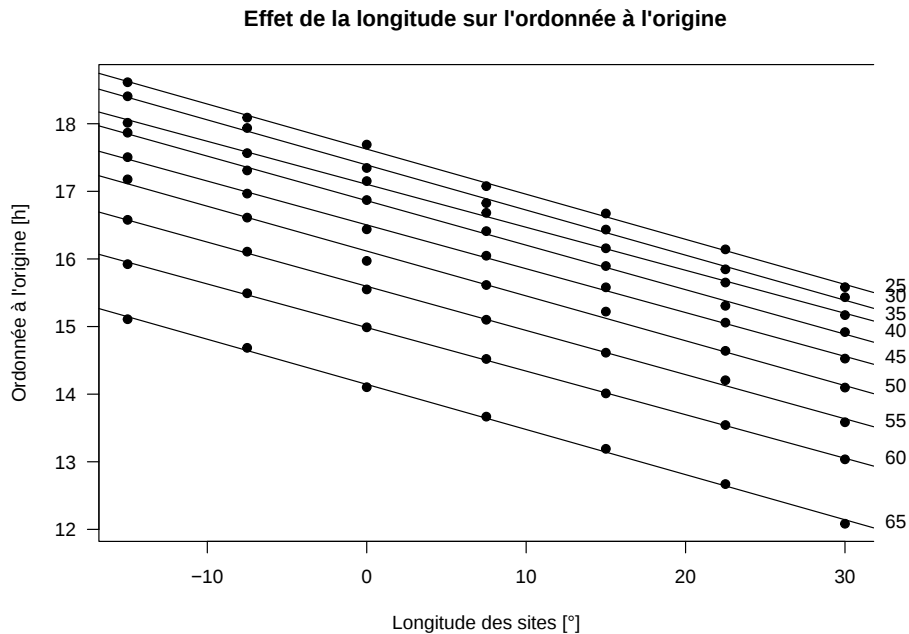
```
with(subset(df, ylat == 45), {
  plot(x, y, xlab = "Date [rang du jour de l'année]", las = 1,
       ylab = "Coucher du soleil UTC [h]", pch = 19,
       main = "Heure de coucher du soleil pour les sites de latitude 45")
  for(site in unique(xlon)){
    ii <- which(xlon == site)
    abline(lm1 <- lm(y[ii]~x[ii]))
    text(par("usr")[2], max(lm1$fitted.values), site, pos = 4, xpd = NA)
  }
  adddebm0is()
})
```

Heure de coucher du soleil pour les sites de latitude 45



DANS le cadre des modèles mixtes, c'est typiquement dans ce genre de situation que l'on sera tenté d'introduire l'ordonnée en effet aléatoire : la pente est commune aux sites, ils diffèrent simplement par cette translation sur l'axe vertical. On vérifie que cette observation est valable pour tous les sites de même latitude :

```
plot.new() ; plot.window(xlim = range(dfp$xlone), ylim = range(dfp$inter))
axis(1) ; axis(2, las = 1) ; box(bty = "c")
title(xlab = "Longitude des sites [°]",
      ylab = "Ordonnée à l'origine [h]",
      main = "Effet de la longitude sur l'ordonnée à l'origine")
for(y in unique(dfp$ylat)){
  with(subset(dfp, ylat == y), {
    points(xlon, inter, pch = 19)
    abline(lm(inter~xlon))
    text(par("usr")[2], min(inter), y, pos = 4, xpd = NA)
  })
}
```



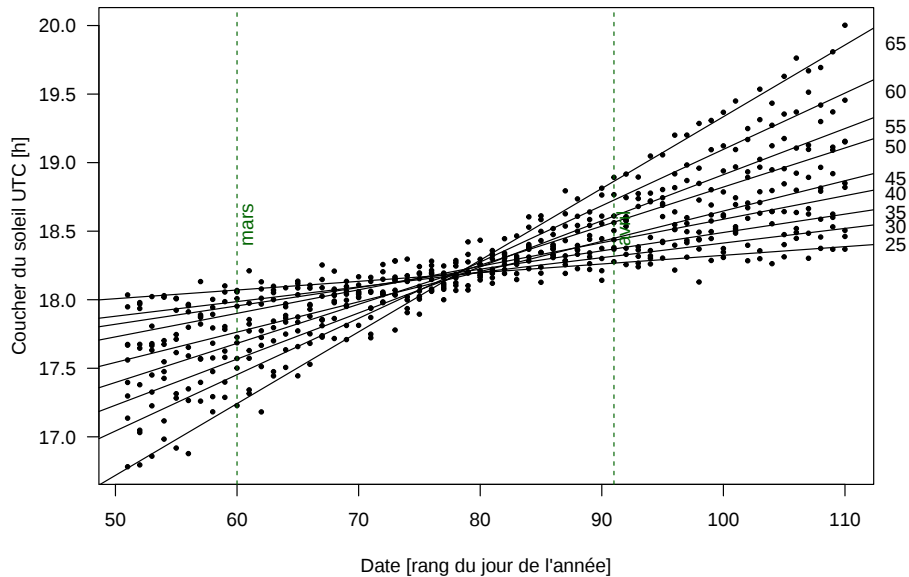
C'EST bien cela : pour tous les sites de même latitude, l'effet de la longitude est du type décalage horaire en translatant toutes les valeurs d'une heure tous les 15° . On visualise également ici un effet de la latitude : pour les sites de même longitude, l'ordonnée à l'origine est plus élevée aux faibles latitudes ; quand on va vers l'équateur les jours sont plus longs et le soleil se couche plus tard.

1.5.2 Effet de la latitude

POUR se faire une première idée, on reprend l'approche pour la longitude mais en pivotant à angle droit : cette fois on représente les données pour tous les sites sur le méridien d'origine.

```
with(subset(df, xlon == 0), {
  plot(x, y, xlab = "Date [rang du jour de l'année]", las = 1,
       ylab = "Coucher du soleil UTC [h]", pch = 19, cex = 0.5,
       main = "Heure de coucher du soleil pour les sites de longitude 0")
  for(site in unique(ylat)){
    ii <- which(ylat == site)
    abline(lm1 <- lm(y[ii]~x[ii]))
    text(par("usr")[2], max(lm1$fitted.values), site, pos = 4, xpd = NA)
  }
  adddebm()
})
```

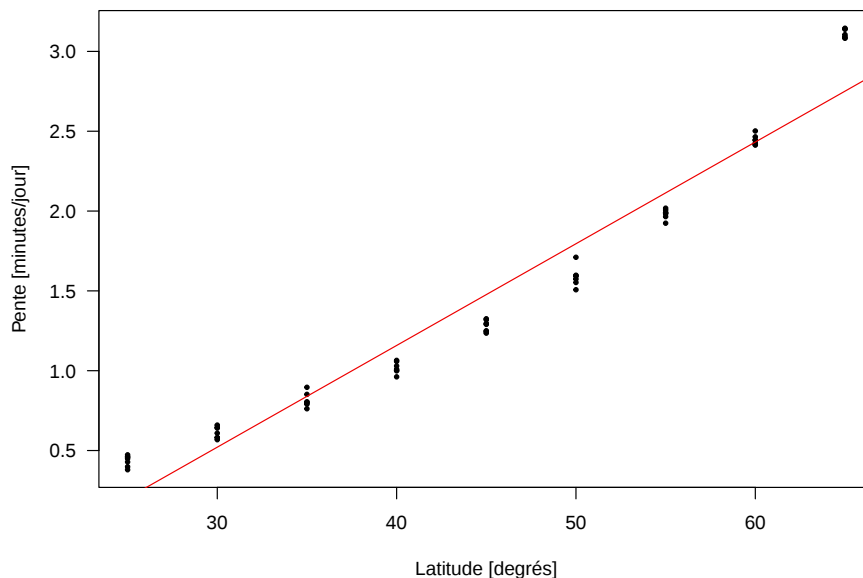
Heure de coucher du soleil pour les sites de longitude 0



La situation est bien différente : la pente augmente avec la latitude, le gain de durée d'ensoleillement par jour est plus rapide au nord qu'au sud, nous avons déjà vu cet effet avec les sites extrêmes sur le méridien d'origine pour comprendre les variations de r^2 avec la latitude (section 1.4 page 9). Dans le cadre des modèles mixtes, c'est typiquement dans ce genre de situation que l'on voudra introduire la pente en effet aléatoire. On vérifie que cette observation est valable pour tous les sites de même latitude :

```
plot(dfp$ylat, dfp$slp, xlab = "Latitude [degrés]", pch = 19,
     ylab = "Pente [minutes/jour]", las = 1, cex = 0.5,
     main = "Augmentation de la pente avec la latitude")
abline(lm(slp~ylat, dfp), col = "red2")
```

Augmentation de la pente avec la latitude



IL y a bien un effet général de la latitude sur les pentes : on passe de 0.5 min.j^{-1} plein Sud à plus de 3 min.j^{-1} plein Nord. La durée du jour augmente plus vite au Nord qu'au Sud, mais c'est bien normal puisque que l'on partait d'une durée du jour bien plus faible. La distribution des résidus du modèle est loin d'être satisfaisante, est-ce à dire que l'approche linéaire a échoué ? Que nenni, bien le rebours !

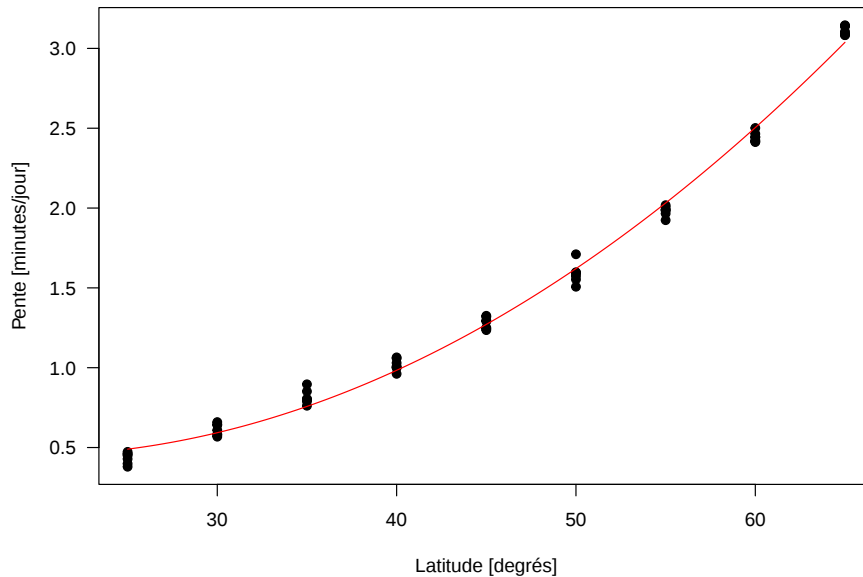
1.6 Modèle linéaire à réponse non linéaire

CETTE partie est parfaitement dispensable pour qui ne s'intéresse qu'aux modèles mixtes et on pourra directement sauter à la section 2 page 22. On profite simplement ici pour tordre le coup à une vieille idée reçue : ce n'est pas parce qu'un modèle est linéaire que sa réponse est forcément linéaire. Par exemple, si on s'intéresse à la relation entre la pente de la droite de régression et la latitude, on constate que le modèle *linéaire* suivant en est un bon résumé.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \quad (5)$$

```
plot(dfp$ylat, dfp$slp, xlab = "Latitude [degrés]", pch = 19,
     ylab = "Pente [minutes/jour]", las = 1,
     main = "Augmentation de la pente avec la latitude")
lmslp2 <- lm(slp~ylat+I(ylat^2), dfp)
xx <- seq(min(latseq), max(latseq), le = 256)
yy <- predict(lmslp2, list(ylat = xx))
lines(xx, yy, col = "red")
```

Augmentation de la pente avec la latitude



POUR être linéaire il faut que toutes les dérivées partielles de la partie déterministe du modèle par rapport aux paramètres ne soient pas fonction des paramètres. Par exemple, pour β_2 dans (eq. 5) tout va bien :

$$\frac{\partial}{\partial \beta_2}(\beta_0 + \beta_1 X + \beta_2 X^2) = X^2 \quad (6)$$

En pratique cela va poser un problème dès qu'un paramètre se retrouve « à l'intérieur » d'une fonction, par exemple :

$$\frac{\partial}{\partial \mu}(e^{\mu X}) = X e^{\mu X} \quad (7)$$

QUAND on fait de la régression linéaire multiple, on a plusieurs variables prédictives $X_1, X_2, \dots, X_p \equiv \mathbf{X}$. En notant $\beta = (\beta_1, \beta_2, \dots, \beta_p)^\top$ le modèle complet avec interactions s'écrit :

$$Y = \beta_0 + \mathbf{X}\beta + \mathbf{X}^\top \mathbf{X} \gamma + \epsilon \quad (8)$$

COMME on vient de voir qu'il y avait une réponse quadratique des coefficients des droites de régression avec la latitude, on ajoute le carré d'icelle dans les variables explicatives.

```
df2 <- df
df2$y1at2 <- df2$y1at^2
```

C'EST une autre façon de comprendre ce qu'est un modèle linéaire : je peux ajouter comme variable explicative n'importe quelle combinaison de fonctions (pas forcément linéaires) des variables de contrôle du moment qu'elle ne

dépende pas des paramètres à estimer : elle est calculée une fois pour toute en amont de toute opération d'estimation des paramètres du modèle linéaire. Ici on a ajouté le carré de la latitude, mais on pourrait, par exemple, ajouter le sinus du cosinus de la somme de la latitude et du rang du jour de l'année tout en restant dans le cadre d'un modèle linéaire. Voyons maintenant ce que donne le modèle complet avec interactions :

```
lmfull <- lm(y ~ .*, df2)
summary(lmfull)

Call:
lm(formula = y ~ . * ., data = df2)
Residuals:
    Min       1Q   Median       3Q      Max
-0.292090 -0.058512 -0.000334  0.061616  0.309485

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.685e+01  1.147e-01  146.915 <2e-16 ***
x             1.507e-02  1.091e-03   13.813 <2e-16 ***
xlon        -6.493e-02  1.334e-03  -48.666 <2e-16 ***
ylat         7.351e-02  6.746e-03   10.897 <2e-16 ***
ylat2       -1.898e-03  1.312e-04  -14.464 <2e-16 ***
x:xlon       -1.291e-05  5.500e-06   -2.348  0.0189 *
x:ylat       -7.856e-04  5.117e-05  -15.351 <2e-16 ***
x:ylat2      2.053e-05  5.641e-07   36.386 <2e-16 ***
xlon:ylat    -2.441e-05  5.908e-05   -0.413  0.6796
xlon:ylat2    1.898e-07  6.513e-07    0.291  0.7707
ylat:ylat2    2.263e-06  9.082e-07    2.492  0.0127 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08785 on 3769 degrees of freedom
Multiple R-squared:  0.9938,    Adjusted R-squared:  0.9938
F-statistic: 6.082e+04 on 10 and 3769 DF,  p-value: < 2.2e-16
```

ON voit que certains termes sont dispensables, on simplifie le modèle sur critère BIC³.

```
lmopt <- step(lmfull, k = log(nrow(df)), trace = 0)
summary(lmopt)

Call:
lm(formula = y ~ x + xlon + ylat + ylat2 + x:ylat + x:ylat2,
    data = df2)
Residuals:
    Min       1Q   Median       3Q      Max
-0.295184 -0.059686  0.000327  0.061531  0.312185

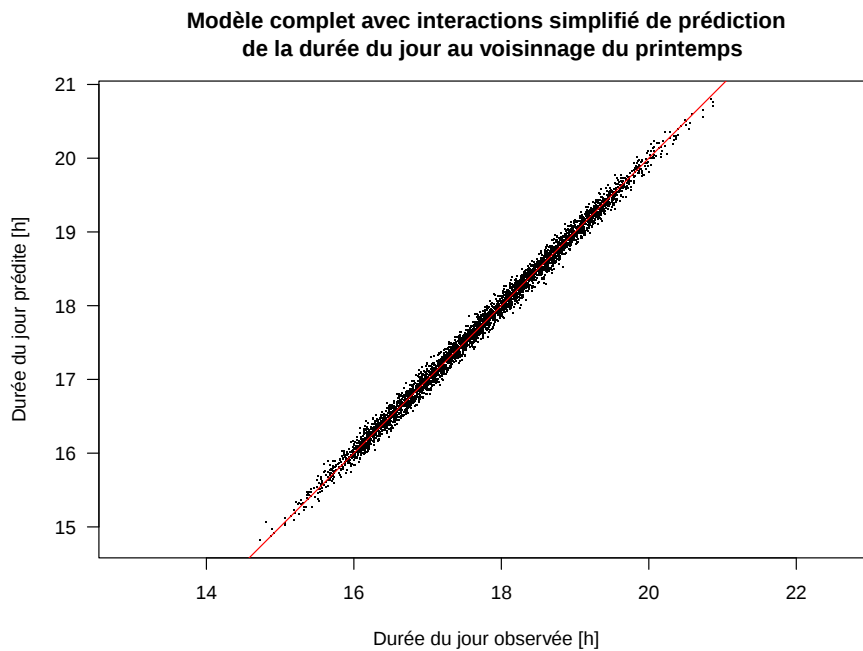
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.704e+01  8.988e-02  189.61 <2e-16 ***
x             1.497e-02  1.091e-03   13.72 <2e-16 ***
xlon        -6.666e-02  9.537e-05 -698.95 <2e-16 ***
ylat         6.025e-02  4.219e-03   14.28 <2e-16 ***
ylat2       -1.591e-03  4.651e-05  -34.20 <2e-16 ***
x:ylat       -7.856e-04  5.123e-05  -15.33 <2e-16 ***
x:ylat2      2.053e-05  5.648e-07   36.34 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08795 on 3773 degrees of freedom
Multiple R-squared:  0.9938,    Adjusted R-squared:  0.9938
F-statistic: 1.011e+05 on 6 and 3773 DF,  p-value: < 2.2e-16
```

ON a donc un modèle linéaire très satisfaisant ($r^2 = 0.99$ pour 7 paramètres résumant 3780 « observations ») avec un RMS de 5.3 minutes. On n'est pas loin des performances du modèle linéaire local.

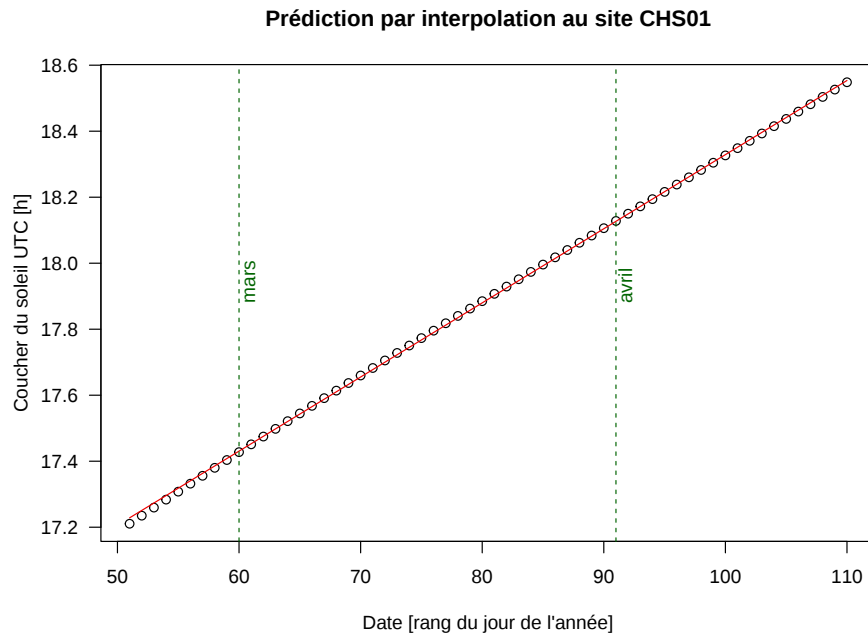
³Ce n'est pas le seul critère possible : AIC [1, 2], TIC [14], BIC [12], AIC_c [9], NIC [11], QAIC et QAIC_c [5], RIC [3], DIC [13], FIC et FRIC [6].

```
plot(df2$y, lmopt$fitted.values, pch = ".", las = 1, asp = 1,
     main = "Modèle complet avec interactions simplifié de prédiction
de la durée du jour au voisinage du printemps",
     xlab = "Durée du jour observée [h]",
     ylab = "Durée du jour prédite [h]")
abline(c(0, 1), col = "red")
```



La très grosse différence avec le modèle linéaire local c'est que nous pouvons utiliser ce modèle pour faire des interpolations ou des extrapolations. On peut faire une interpolation pour voir comment se comporte le modèle pour le site CHS01 de la forêt domaniale de SEILLON (en FRANCE près de BOURG-EN-BRESSE) :

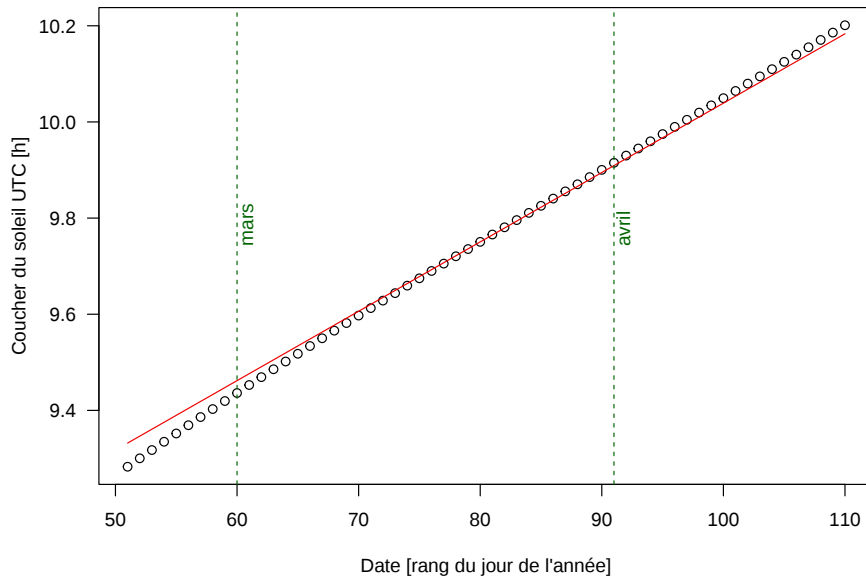
```
predlm <- function(xx, yy, mylm, ...){
  pred <- predict(mylm, list(x = yday(tobs),
                             xlon = rep(xx , nt),
                             ylat = rep(yy , nt),
                             ylat2 = rep(yy, nt)^2))
  obs <- cds(tobs, yy, xx, sd = 0)
  plot(yday(tobs), obs, xlab = "Date [rang du jour de l'année]", las = 1,
       ylab = "Coucher du soleil UTC [h]", ...)
  lines(yday(tobs), pred, col = "red2")
  adddebmois()
}
yy <- 46 + 10/60 + 17/3600 ; xx <- 5 + 14/60 + 22/3600 # CHS01
predlm(xx, yy, lmopt, main = "Prédiction par interpolation au site CHS01")
```



LES prédictions du modèle (droite rouge) sont très bonnes puisque l'on est à une épaisseur de trait de la réalité. On peut essayer d'utiliser le modèle pour extrapoler, pour ce qui est des longitudes on se doute que ça ne devrait pas être trop mal puisque c'est essentiellement une question de décalage horaire. Voyons ce qu'il en est à SÉOUL :

```
yy <- 37 + 33/60 + 57/3600 ; xx <- 126 + 58/60 + 41/3600 # Séoul
predlm(xx, yy, lmopt, main = "Prédiction par par extrapolation à Séoul")
```

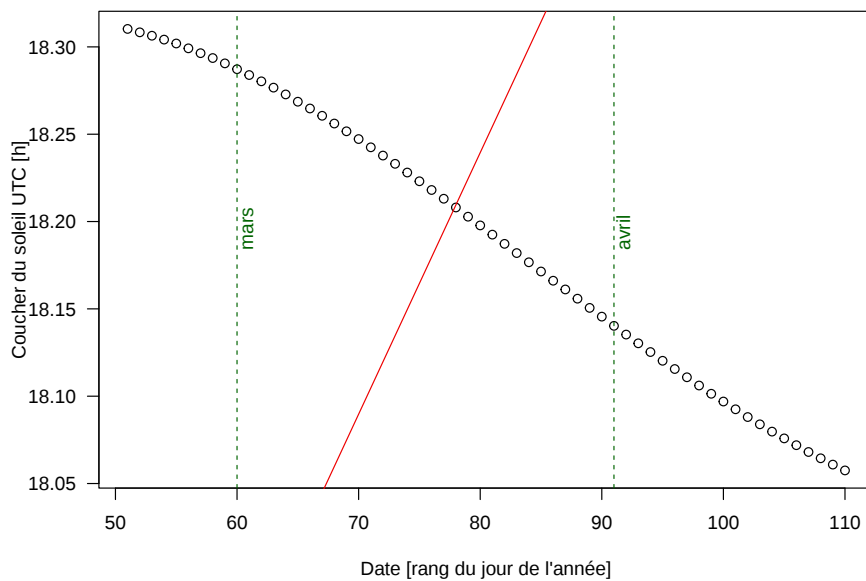
Prédiction par extrapolation à Séoul



L'EXTRAPOLATION en longitude n'est pas trop mauvaise. En revanche, si on extrapole pour les latitudes en deçà du tropique du CANCER, on va complètement sortir du domaine de validité du modèle et avoir des prédictions complètement aberrantes :

```
yy <- 0 ; xx <- 0 # méridien d'origine à l'équateur
predlm(xx, yy, lmopt, main = "Prédiction par extrapolation à l'équateur")
```

Prédiction par extrapolation à l'équateur



C'EST encore pire au pôle Nord puisque le modèle prédit sans broncher les heures de coucher du soleil au printemps alors qu'il ne s'y couche qu'une fois par an à l'équinoxe d'automne.

2 Le jeu de données dégradées

2.1 Dégradation totale de l'information

POUR illustrer les modèles mixtes, nous allons supposer que nous sommes incapables de mesurer la latitude et la longitude des sites. Notre variable `site` est maintenant une simple variable qualitative (non ordonnée) qui porte le nom de chaque site.

```
dfq <- df[, c("y", "x")]
dfq$site <- as.factor(paste0(df$xlon, " lon ", df$ylat, " lat"))
head(dfq)
```

	y	x	site
1	18.91891	51	-15 lon 25 lat
2	18.99614	52	-15 lon 25 lat
3	18.92064	53	-15 lon 25 lat
4	19.13294	54	-15 lon 25 lat
5	19.03690	55	-15 lon 25 lat
6	18.95024	56	-15 lon 25 lat

NOTONS que cette dégradation de l'information est extrêmement brutale. Même si je ne suis plus capable de mesurer la latitude et la longitude des sites, à l'aide d'une simple boussole je suis quand même capable de voir que quand je me rends de tel à tel site je navigue, par exemple, plein Nord.

2.2 Dégradation partielle de l'information

DANS cette section, également dispensable pour qui est pressé d'en arriver aux modèles mixtes, on sait se servir d'une boussole et on remplace la valeur des latitudes et des longitudes par leur rang.

```
dfr <- df[, c("y", "x")]
dfr$rxlon <- unclass(ordered(df$xlon))
dfr$rylat <- unclass(ordered(df$ylat))
head(dfr)
```

	y	x	rxlon	rylat
1	18.91891	51	1	1
2	18.99614	52	1	1
3	18.92064	53	1	1
4	19.13294	54	1	1
5	19.03690	55	1	1
6	18.95024	56	1	1

ON fait un modèle prédictif analogue à celui de la section 1.6 page 16, simplement on travaille ici avec les rangs :

```
dfr$rylat2 <- dfr$rylat^2
lmrfull <- lm(y ~ .*, dfr)
lmropt <- stepAIC(lmrfull, k = log(nrow(dfr)), trace = 0)
summary(lmropt)
```

Call:

```
lm(formula = y ~ x + rxlon + rylat2 + x:rylat2, data = dfr)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.29353	-0.05923	-0.00040	0.06148	0.31575

Coefficients:

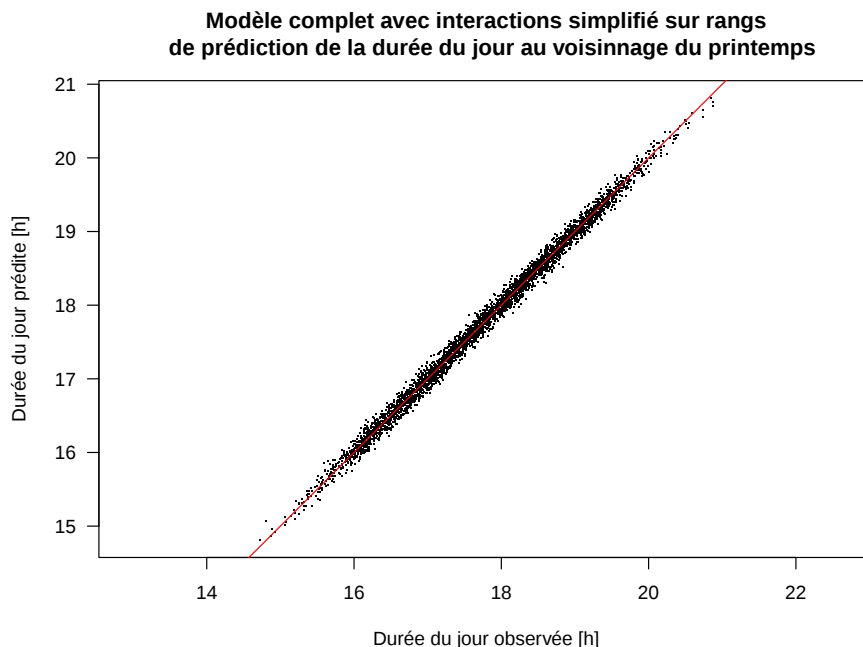
	Estimate	Std. Error	t value	Pr(> t)
--	----------	------------	---------	----------

```
(Intercept) 1.908e+01 1.098e-02 1736.72 <2e-16 ***
x           7.825e-03 1.288e-04 60.76 <2e-16 ***
rxlon      -4.999e-01 7.153e-04 -698.89 <2e-16 ***
rylat2     -4.138e-02 2.569e-04 -161.05 <2e-16 ***
x:rylat2    5.301e-04 3.120e-06 169.87 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08796 on 3775 degrees of freedom
Multiple R-squared:  0.9938,    Adjusted R-squared:  0.9938
F-statistic: 1.517e+05 on 4 and 3775 DF,  p-value: < 2.2e-16
```

Le modèle obtenu est aussi bon que celui de la section 1.6 page 16, voire meilleur puisqu'il comporte moins de paramètres. Graphiquement :

```
plot(dfr$y, lmropt$fitted.values, pch = ".", las = 1, asp = 1,
     main = "Modèle complet avec interactions simplifié sur rangs
de prédiction de la durée du jour au voisinage du printemps",
     xlab = "Durée du jour observée [h]",
     ylab = "Durée du jour prédite [h]")
abline(c(0, 1), col = "red")
```



En conclusion, si les sites sont caractérisés par une ou plusieurs variables qualitatives *ordonnées*, il serait dommage de perdre cette information en fonçant tête baissée dans les modèles mixtes.

2.3 Modèle linéaire global

Si on représente les données tous sites confondus, le résultat est assez décevant dans le sens où notre modèle linéaire global n'explique qu'une faible fraction ($r^2 = 0.15$) de la variabilité de l'heure de coucher du soleil.

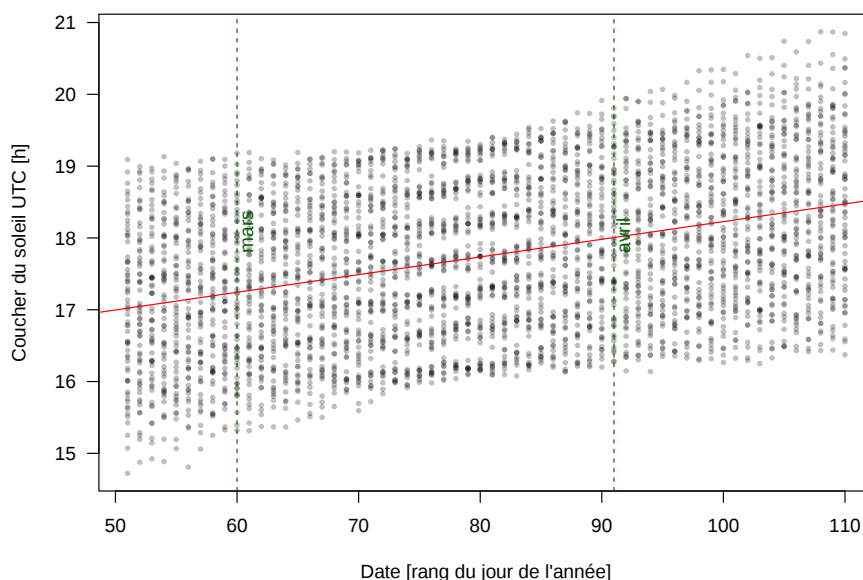
```
plot(df$x, df$y, xlab = "Date [rang du jour de l'année]", las = 1,
     ylab = "Coucher du soleil UTC [h]", pch = 19, cex = 0.5, col = rgb(0,0,0,0.25),
     main = "Heure de coucher du soleil tous sites confondus")
abline(lmg <- lm(df$y~df$x), col = "red")
adddebmois()
summary(lmg)
```

```
Call:
lm(formula = df$y ~ df$x)
Residuals:
    Min       1Q   Median       3Q      Max
-2.33643 -0.88836  0.00084  0.88769  2.44580

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.577e+01  7.993e-02  197.26  <2e-16 ***
df$x        2.461e-02  9.707e-04   25.35  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.034 on 3778 degrees of freedom
Multiple R-squared:  0.1454,    Adjusted R-squared:  0.1452
F-statistic: 642.8 on 1 and 3778 DF,  p-value: < 2.2e-16
```

Heure de coucher du soleil tous sites confondus



ON peut quand même en tirer quelques enseignements : globalement, tous sites confondus, il y a une augmentation de l'heure de coucher du soleil au voisinage du printemps. C'est bien normal : tous nos sites sont dans l'hémisphère nord entre le tropique du CANCER et le cercle polaire. La pente est de 1.5 minutes par jour, soit une valeur proche de celle observée au centre du monde.

EN plissant un peu les yeux au voisinage de l'équinoxe on voit un phénomène que l'on appellerait une lézarde (ou une rivière) en typographie. Nous verrons plus loin, section 3.4.1 page 32, la cause de cette rupture du gris typographique.

CE qui est clair, à ce stade, c'est que la magnifique droite de régression obtenue au centre du monde que nous avons exhibée (section 1.2 page 3) pour broser nos financeurs dans le sens du poil ne se généralise absolument pas. Pour rendre compte des observations il va falloir individualiser les modèles site par site. C'est tout l'objet des modèles mixtes.

3 Modèles linéaires mixtes

3.1 Syntaxe

NOUS utiliserons ici la syntaxe de la fonction `lmer()` du paquet `lme4` [4]. La partie fixe se note comme pour les modèles linéaires habituels, la partie aléatoire suit entre parenthèses. Il y a parfois plusieurs façons de noter la même chose, une seule a été reprise ici. On n'a listé ici que quelques exemples, pour plus de détails voir la FAQ⁴.

Formule	Sens
<code>(1 site)</code>	Ordonnée aléatoire par site
<code>(0+x site)</code>	Pente aléatoire par site
<code>(x site)</code>	Pente aléatoire par site avec ordonnées corrélées
<code>(1 site) + (0+x site)</code>	Pente aléatoire par site avec ordonnées non corrélées

3.2 Ordonnées en effet aléatoire

C'EST sans doute l'application la plus courante des modèles mixtes : on suppose que la pente est la même pour tous les sites et qu'ils ne diffèrent que par l'ordonnée à l'origine. On suppose déjà que cela ne va pas très bien marcher ici, mais cela permet de bien voir ce qui est fait dans ce cas et d'examiner les outils qui nous permettent de poser le diagnostic d'un « mauvais » modèle. Par rapport au modèle linéaire global on introduit un peu de flexibilité en rouge, ${}^s\beta_0$, en autorisant donc l'ordonnée à l'origine à varier pour chaque site s :

$$\forall s \in \{1, 2, \dots, n\} : {}^sY = {}^s\beta_0 + \beta_1 {}^sX + \epsilon \quad (9)$$

QUAND on utilise des modèles mixtes on suppose que un ou plusieurs paramètres de la relation fixe sont issus d'une distribution que l'on suppose généralement normale. Ici on suppose que les pentes à l'origine sont issues d'une distribution normale :

$$\beta_0 \sim \mathcal{N}(\mu_{\beta_0}, \sigma_{\beta_0}) \quad (10)$$

N'ALLEZ pas croire que par rapport au modèle linéaire global nous n'aurions ajouté qu'un seul paramètre en substituant à β_0 les deux paramètres populationnels μ_{β_0} et σ_{β_0} . Non, chaque site aura bien son ordonnée à l'origine individuelle et nous sommes passés de 2 paramètres à $n + 1$ paramètres. Le gros intérêt de la distribution normale est qu'elle va induire un effet de contraction (*shrinkage*) ou encore dit de régularisation sur la distribution des ordonnées à l'origine. Ceci va être intéressant en cas de plan d'échantillonnage déséquilibré : supposons, par exemple, que les sites en haute mer soient beaucoup moins bien documentés à cause d'un problème de mal de mer⁵ rendant les observations souvent impossibles à réaliser. L'estimation de l'ordonnée à l'origine sera alors

⁴<https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>

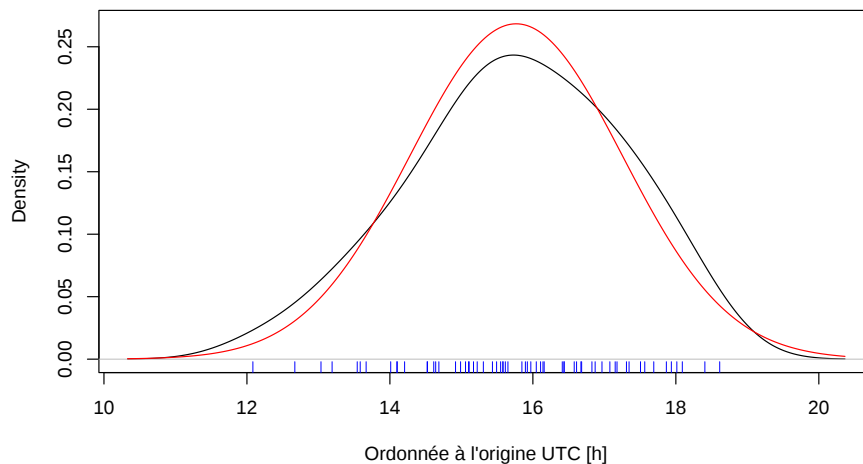
⁵Je n'arrive pas retrouver la source de cette citation du manuel des GLÉNANS disant, de mémoire : « on reconnaît que le capitaine a le mal de mer à ce que ses ordres passent de l'impératif au conditionnel ». Si quelqu'un a la référence, je suis preneur.

beaucoup plus fluctuante pour ces sites mal renseignés, et l'effet stabilisateur des sites en terre ferme sera alors le bienvenu.

DANS notre cas, la distribution de l'ordonnée à l'origine des régressions linéaires par site n'est clairement pas normale (courbe en rouge ci-après), mais ce n'est pas non plus un résumé très éloigné de la réalité.

```
mydst <- function(x, ...){
  dst <- density(x)
  xx <- seq(min(dst$x), max(dst$x), le = 256)
  yy <- dnorm(xx, mean(x), sd(x))
  plot(dst <- density(x), ylim = c(0, max(dst$y, yy)), ...)
  lines(xx, yy, col = "red")
  rug(x, col = "blue")
}
mydst(df$inter, main = "Distribution des ordonnées à l'origine",
      xlab = "Ordonnée à l'origine UTC [h]")
```

Distribution des ordonnées à l'origine



ON utilise la fonction `lmer()` du paquet `lmer4` [?] pour estimer les paramètres du modèle avec la syntaxe détaillée dans la table de la section 3.1 page 25.

```
library(lme4)
lmer1 <- lmer(y~x + (1|site), dfq)
```

ON définit une petite fonction `showme()` pour représenter les données et l'ajustement du modèle sur le méridien d'origine pour les deux latitudes les plus extrêmes et au centre du monde.

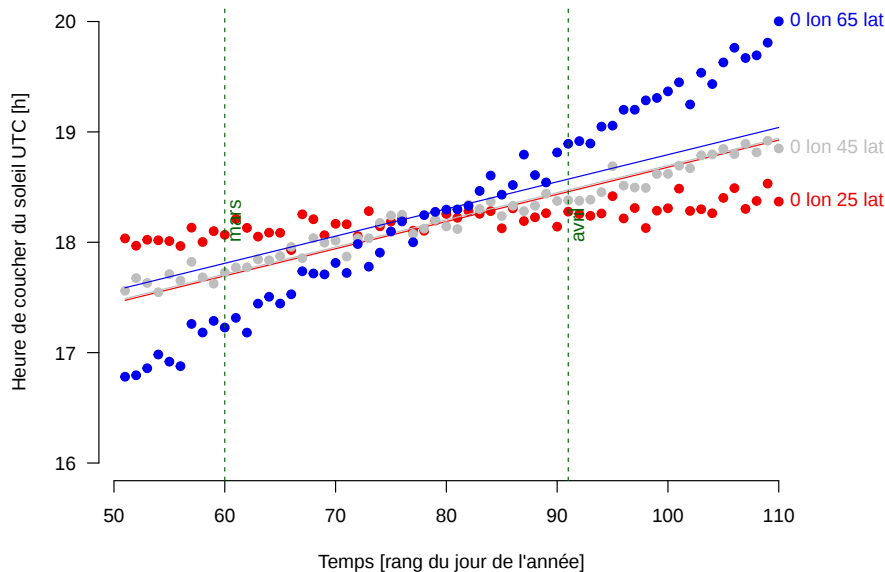
```
showme <- function(lmer){
  par(mar = c(5, 4, 4, 5) + 0.1)
  plot.new() ; plot.window(xlim = range(dfq$x), ylim = c(16, 20))
  dfq$ytheo <- predict(lmer)
  addsite <- function(the_site, ...){
    with(subset(dfq, site == the_site), {
      points(x, y, pch = 19, ...)
      points(x, ytheo, type = "l", ...)
      text(110, y[length(y)], the_site, xpd = NA, pos = 4, ...)
    })
  }
  addsite("0 lon 25 lat", col = "red2")
  addsite("0 lon 45 lat", col = "grey")
  addsite("0 lon 65 lat", col = "blue2")
  axis(1) ; axis(2, las = 1)
```

```

title(xlab = "Temps [rang du jour de l'année]")
title(ylab = "Heure de coucher du soleil UTC [h]")
title(main = paste("Modèle mixte", deparse(substitute(lmer)), "sur le méridien d'origine"))
adddebm()
}
showme(lmer1)

```

Modèle mixte lmer1 sur le méridien d'origine



L'AJUSTEMENT est à l'évidence très mauvais, sauf au centre du monde. La contrainte d'avoir une pente commune à tous nos sites ne permet pas de rendre compte de la réalité. Il va falloir mettre la pente en effet aléatoire. Est-ce à dire que ce modèle mixte ne nous a rien appris ? Non, regardons les ajustements pour tous les sites avec la fonction `showall()` définie ci-après :

```

showall <- function(lmer, xlim = NULL){
  par(mar = c(5, 4, 4, 5) + 0.1)
  if(is.null(xlim)) xlim <- range(dfq$x)
  plot.new() ; plot.window(xlim = xlim, ylim = range(dfq$y))
  cfs <- coef(lmer)$site
  points(dfq$x, dfq$y, pch = ".")
  for(i in seq_len(nrow(cfs))) abline(c(cfs[i,1], cfs[i,2]))
  axis(1) ; axis(2, las = 1)
  title(xlab = "Temps [rang du jour de l'année]")
  title(ylab = "Heure de coucher du soleil UTC [h]")
  title(main = paste("Modèle mixte", deparse(substitute(lmer)), "pour tous les sites"))
  adddebm()
}
showall(lmer1)

```



ON voit très clairement apparaître 7 groupes de sites qui diffèrent par une translation verticale de 30 minutes. Ce sont les sites de même longitude. La fonction `model_performance()` du paquet `performance` [10] donne un résumé des « performances » d'un modèle mixte :

```
library(performance)
performance(lmer1)
# Indices of model performance
AIC | AICc | BIC | R2 (cond.) | R2 (marg.) | ICC | RMSE | Sigma
-----|-----|-----|-----|-----|-----|-----|-----
1005.8 | 1005.9 | 1030.8 | 0.946 | 0.144 | 0.937 | 0.258 | 0.260
```

Le premier indice à notre disposition pour juger de la qualité du modèle est sans doute le plus « opérationnel » : le RMSE (*Root Mean Square Estimate*) nous donne une idée de la précision des prédictions du modèle, c'est la racine carrée de la moyenne des carrés des résidus entre les observations et les prédictions du modèle, et avec sa valeur d'un quart d'heure, soit 15 minutes, on est bien loin des 5 minutes attendues :

```
sqrt(mean(residuals(lmer1)^2))
[1] 0.2581638
60*sqrt(mean(residuals(lmer1)^2))
[1] 15.48983
```

Le deuxième indice, r_c^2 , le r^2 conditionnel correspond au classique r^2 des modèles linéaires : c'est la fraction de la variance totale expliquée par le modèle, effets fixes et aléatoires confondus :

$$r_c^2 = \frac{\sigma_f^2 + \sigma_r^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\epsilon^2}$$

```
1 - var(residuals(lmer1))/var(getData(lmer1)$y)
[1] 0.9466513
```

NOUS avons donc 95 % de la variance totale qui est prise en compte par le modèle, comme quoi on peut avoir un bon r^2 avec un mauvais modèle. Si le r^2 est bon c'est parce que la durée du jour augmente partout au printemps dans l'hémisphère Nord, il n'aurait pas été aussi bon si on avait inclus des sites dans l'hémisphère Sud où la durée du jour décroît au printemps.

LE troisième indice, r_m^2 , le r^2 marginal correspond à la fraction de la variance expliquée uniquement par les effets fixes du modèle mixte :

$$r_m^2 = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_r^2 + \sigma_\epsilon^2}$$

```
1 - var(residuals(lmg))/var(dfq$y)
[1] 0.1454033
```

ICI la valeur du r_c^2 (0.948) est très améliorée par rapport à celle de la valeur du r_m^2 (0.14), c'est donc bien qu'il y a un intérêt à mettre l'ordonnée à l'origine en effet aléatoire. Cependant, le RMSE nous dit que nous sommes encore loin du compte.

ON inspecte la valeurs des coefficients pour vérifier que l'ordonnée à l'origine est variable d'un site à l'autre alors que la pente est commune à tous les sites :

```
head(coef(lmer1)$site)
      (Intercept)      x
-15 lon 25 lat    17.23403 0.0246105
-15 lon 30 lat    17.23987 0.0246105
-15 lon 35 lat    17.23333 0.0246105
-15 lon 40 lat    17.23818 0.0246105
-15 lon 45 lat    17.26210 0.0246105
-15 lon 50 lat    17.27724 0.0246105
tail(coef(lmer1)$site)
      (Intercept)      x
 7.5 lon 40 lat    15.77126 0.0246105
 7.5 lon 45 lat    15.73546 0.0246105
 7.5 lon 50 lat    15.76902 0.0246105
 7.5 lon 55 lat    15.78038 0.0246105
 7.5 lon 60 lat    15.79282 0.0246105
 7.5 lon 65 lat    15.83748 0.0246105
```

3.3 Pentés en effet aléatoire

C'EST le pendant du modèle mixte avec l'ordonnée à l'origine en effet aléatoire : cette fois c'est l'ordonné à l'origine β_0 qui est commune à tous les sites et les pentes $^s\beta_1$ qui varient d'un site à l'autre.

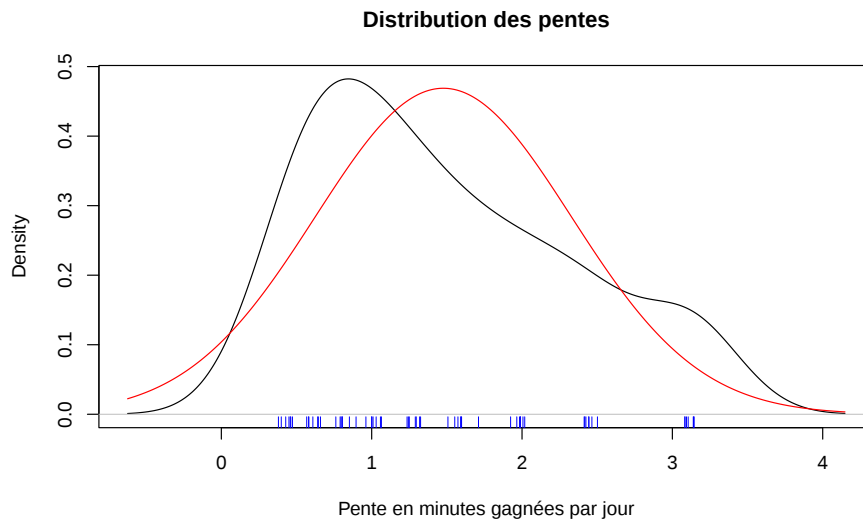
$$\forall s \in \{1, 2, \dots, n\} : ^sY = \beta_0 + ^s\beta_1 ^sX + \epsilon \quad (11)$$

LES pentes suivent une loi normale avec les paramètres populationnels μ_{β_1} et σ_{β_1} . Il y $n + 1$ paramètres en tout dans ce modèle mixte.

$$\beta_1 \sim \mathcal{N}(\mu_{\beta_1}, \sigma_{\beta_1}) \quad (12)$$

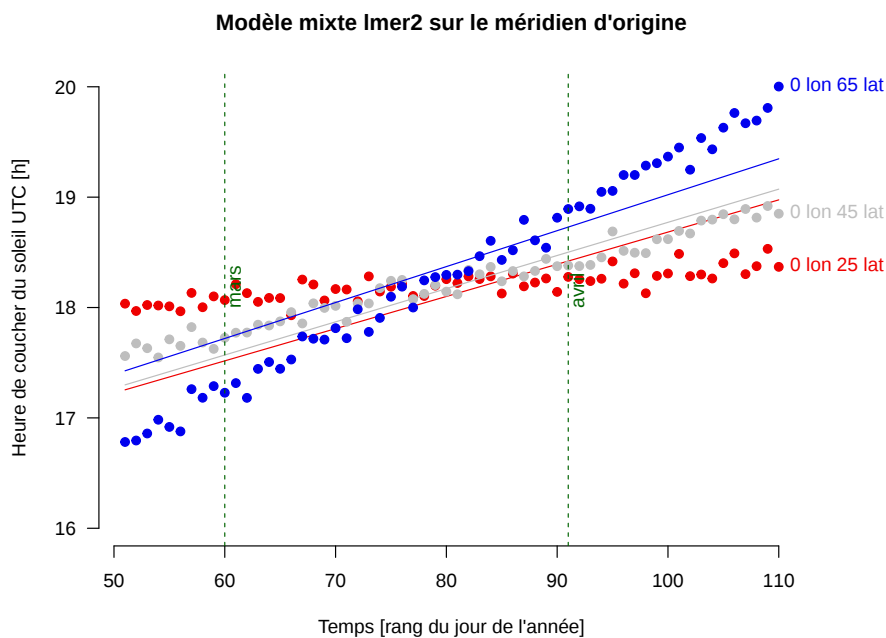
LA distribution des pentes du modèle linéaire local montre que l'on s'éloigne d'une distribution normale : la distribution est asymétrique à droite.

```
mydst(df$slp, main = "Distribution des pentes",
      xlab = "Pente en minutes gagnées par jour")
```



L'AJUSTEMENT du modèle sur nos trois sites choisis sur le méridien d'origine n'est pas bon, sauf au centre du monde :

```
lmer2 <- lmer(y ~ x + (0 + x | site), dfq)
showme(lmer2)
```



L'AJUSTEMENT du modèle pour tous les sites n'est pas non plus très bon, on note que l'on retrouve 7 groupes de sites qui ont une pente assez voisine.

```
showall(lmer2)
```



```
performance(lmer2)
# Indices of model performance
AIC | AICc | BIC | R2 (cond.) | R2 (marg.) | ICC | RMSE | Sigma
-----|-----|-----|-----|-----|-----|-----|-----
2638.4 | 2638.4 | 2663.3 | 0.917 | 0.144 | 0.903 | 0.322 | 0.324
```

NOTRE r_c^2 de 0.917 pourrait sembler être satisfaisant mais on voit bien qu'avec un RMSE de 0.322 (soit un vingtaine de minutes) on est bien loin du compte. On vérifie enfin qu'ici c'est bien l'ordonnée à l'origine qui est commune à tous les sites :

```
head(coef(lmer2)$site)
      (Intercept)      x
-15 lon 25 lat  15.76641 0.04126778
-15 lon 30 lat  15.76641 0.04145411
-15 lon 35 lat  15.76641 0.04158802
-15 lon 40 lat  15.76641 0.04172865
-15 lon 45 lat  15.76641 0.04222397
-15 lon 50 lat  15.76641 0.04259206
tail(coef(lmer2)$site)
      (Intercept)      x
 7.5 lon 40 lat  15.76641 0.02431712
 7.5 lon 45 lat  15.76641 0.02407218
 7.5 lon 50 lat  15.76641 0.02472627
 7.5 lon 55 lat  15.76641 0.02514923
 7.5 lon 60 lat  15.76641 0.02562164
 7.5 lon 65 lat  15.76641 0.02664407
```

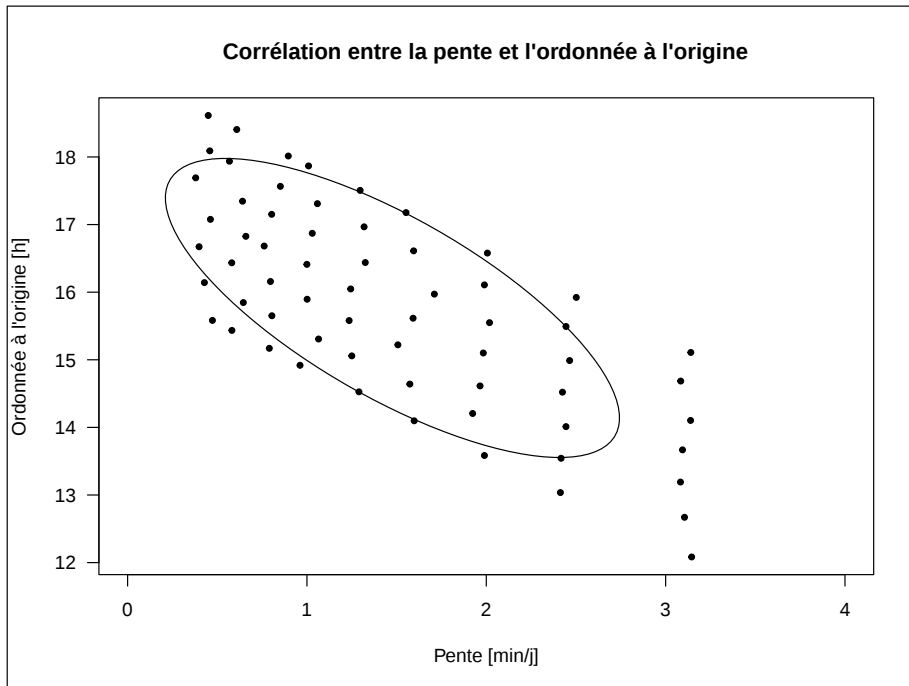
3.4 Pentés et ordonnées en effet aléatoire

CETTE fois se sont à la fois l'ordonnée à l'origine et la pente de la droite de régression qui sont libres de varier d'un site à l'autre. On a donc un modèle mixte avec $2n$ paramètres.

$$\forall s \in \{1, 2, \dots, n\} : {}^s Y = {}^s \beta_0 + {}^s \beta_1 {}^s X + \epsilon \quad (13)$$

IL a deux façons de procéder selon que l'on considère que les pentes et les ordonnées à l'origine sont corrélées ou non. Dans notre cas, il y a une corrélation modérée ($r^2 = 0.54$) entre ces deux paramètres du modèle local.

```
plot(dfp$slp, dfp$inter, pch = 19, cex = 0.5, las = 1,
     xlab = "Pente [min/j]", ylab = "Ordonnée à l'origine [h]",
     main = "Corrélation entre la pente et l'ordonnée à l'origine",
     xlim = c(0, 4))
library(ade4)
s.class(dfp[,c("slp", "inter")], gl(1,nrow(dfp)), add.plot = TRUE, axesell = FALSE,
        clabel = 0, cstar = 0)
```



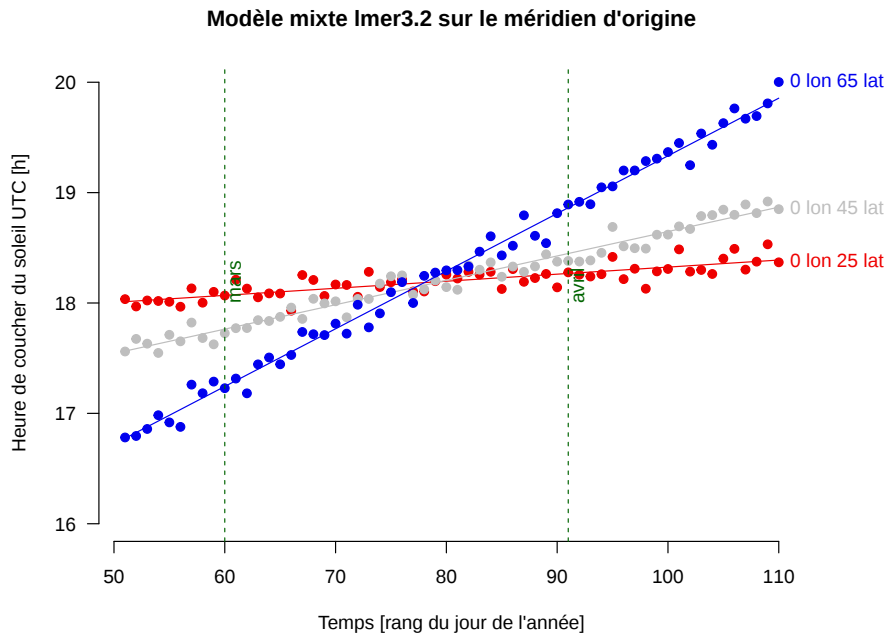
3.4.1 Pentés et ordonnées corrélées

ON suppose que la pente et l'ordonnée à l'origine sont issues d'une distribution normale bivariée :

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \mu_{\beta_0} \\ \mu_{\beta_1} \end{pmatrix}, \begin{pmatrix} \sigma_{\beta_0}^2 & \sigma_{\beta_0 \beta_1} \\ \sigma_{\beta_0 \beta_1} & \sigma_{\beta_1}^2 \end{pmatrix} \right) \quad (14)$$

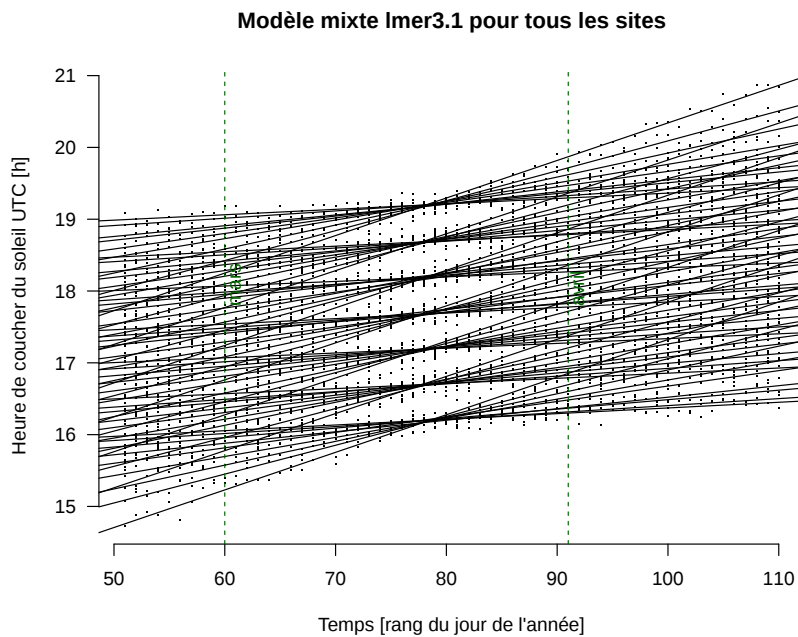
L'AJUSTEMENT du modèle sur nos trois sites de référence est (enfin !) satisfaisant : il n'y a pas qu'au centre du monde que tout se passe bien :

```
lmer3.1 <- lmer(y ~ x + (x | site), dfq) # warning
# from ?lme4::troubleshooting :
# refitting the parameters starting at the current estimates: getting consistent
# results (with no warning) suggests a false positive
lmer3.1 <- update(lmer3.1, start = getME(lmer3.1, "theta"))
showme(lmer3.1)
```



L'AJUSTEMENT du modèle tous sites confondus est également beaucoup plus satisfaisant. On comprend ici l'origine des lézardes typographiques que nous avons remarquées à la section 2.3 page 23 : tous les modèles se croisent à l'équinoxe du printemps, d'où une distribution plus ponctuelles des points à ce niveau.

```
showall(lmer3.1)
```



```
performance(lmer3.1)
# Indices of model performance
AIC      |      AICc |      BIC | R2 (cond.) | R2 (marg.) | ICC | RMSE | Sigma
-----|-----|-----|-----|-----|-----|-----|-----|-----
-6769.8 | -6769.8 | -6732.4 | 0.994 | 0.143 | 0.993 | 0.085 | 0.087
```

NOUS avons enfin un RMSE digne de ce nom de 0.085 h, soit 5.1 min, conforme à l'amplitude du bruit gaussien que nous avons introduit. On vérifie que les coefficients sont variables d'un site à l'autre :

```
head(coef(lmer3.1)$site)
      (Intercept)      x
-15 lon 25 lat    18.61021 0.00753302
-15 lon 30 lat    18.40308 0.01017866
-15 lon 35 lat    18.01191 0.01495673
-15 lon 40 lat    17.86559 0.01683463
-15 lon 45 lat    17.50461 0.02161632
-15 lon 50 lat    17.17691 0.02587542

tail(coef(lmer3.1)$site)
      (Intercept)      x
 7.5 lon 40 lat    16.40982 0.01667809
 7.5 lon 45 lat    16.04689 0.02074147
 7.5 lon 50 lat    15.61475 0.02652693
 7.5 lon 55 lat    15.10191 0.03303887
 7.5 lon 60 lat    14.52336 0.04038054
 7.5 lon 65 lat    13.67198 0.05151209
```

3.4.2 Pentés et ordonnées non corrélées

ON suppose ici que la covariance entre la pente et l'ordonnée à l'origine est nulle :

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \mu_{\beta_0} \\ \mu_{\beta_1} \end{pmatrix}, \begin{pmatrix} \sigma_{\beta_0}^2 & 0 \\ 0 & \sigma_{\beta_1}^2 \end{pmatrix} \right) \quad (15)$$

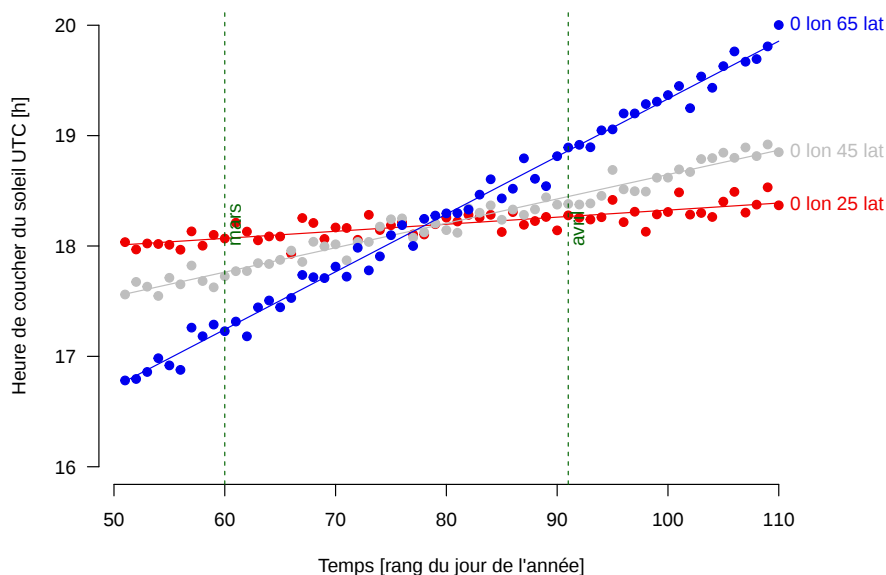
Autrement dit, chaque paramètre suit sa propre loi normale.

$$\beta_0 \sim \mathcal{N}(\mu_{\beta_0}, \sigma_{\beta_0}) \quad ; \quad \beta_1 \sim \mathcal{N}(\mu_{\beta_1}, \sigma_{\beta_1}) \quad (16)$$

DANS ce dernier modèle, la pente et l'ordonnée sont libres de varier d'un site à l'autre. C'est ce qui se rapproche le plus de l'ajustement site par site, mais n'est pas tout à fait la même chose comme nous allons le voir.

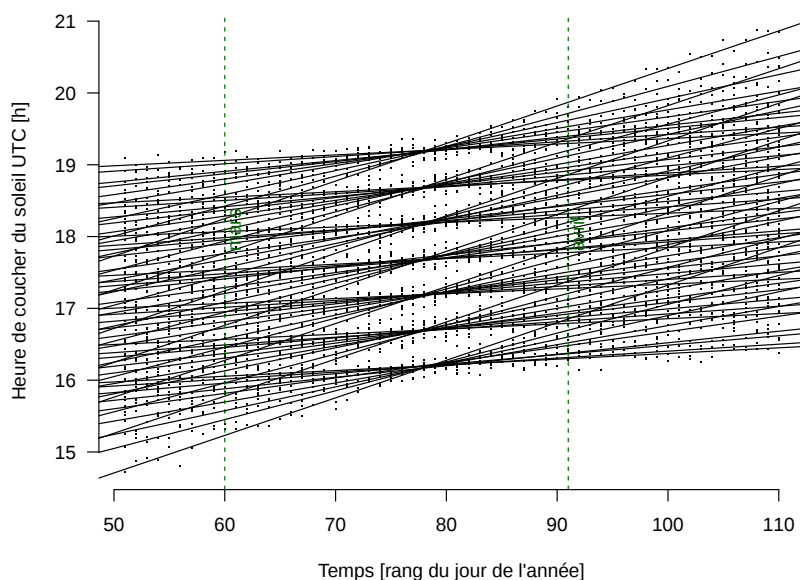
```
lmer3.2 <- lmer(y ~ x + (1 | site) + (0 + x | site), dfq)
showme(lmer3.2)
```

Modèle mixte lmer3.2 sur le méridien d'origine



```
showall(lmer3.2)
```

Modèle mixte lmer3.2 pour tous les sites



```
performance(lmer3.2)
```

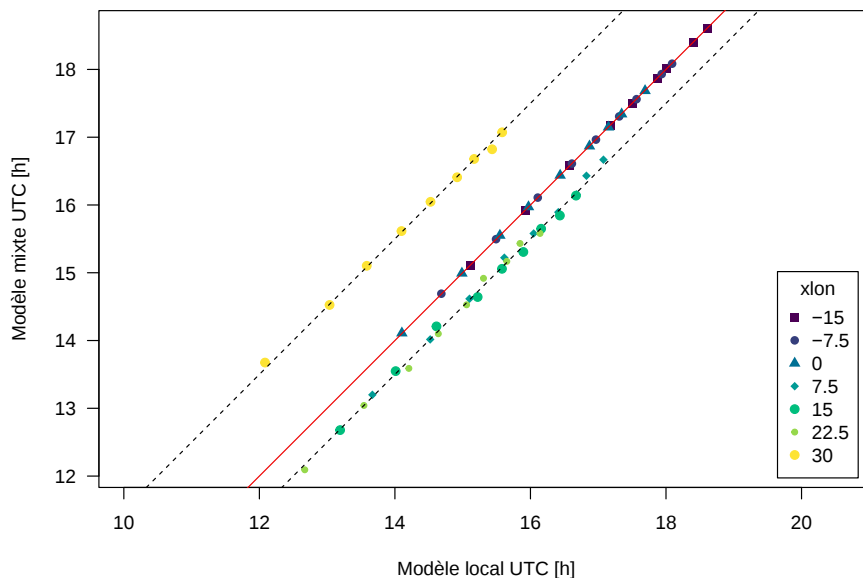
```
# Indices of model performance
AIC | AICc | BIC | R2 (cond.) | R2 (marg.) | ICC | RMSE | Sigma
-----|-----|-----|-----|-----|-----|-----|-----
-6723.7 | -6723.7 | -6692.5 | 0.997 | 0.076 | 0.997 | 0.085 | 0.087
```

```
head(coef(lmer3.2)$site)
      (Intercept)      x
-15 lon 25 lat    18.60691 0.007574365
-15 lon 30 lat    18.40004 0.010216836
-15 lon 35 lat    18.00937 0.014988942
-15 lon 40 lat    17.86325 0.016864596
-15 lon 45 lat    17.50273 0.021640751
-15 lon 50 lat    17.17545 0.025894823

tail(coef(lmer3.2)$site)
      (Intercept)      x
 7.5 lon 40 lat    16.40900 0.01668790
 7.5 lon 45 lat    16.04652 0.02074580
 7.5 lon 50 lat    15.61494 0.02652461
 7.5 lon 55 lat    15.10277 0.03302871
 7.5 lon 60 lat    14.52496 0.04036153
 7.5 lon 65 lat    13.67468 0.05148001

myfac <- as.factor(dfp$xlone) ; nf <- length(levels(myfac))
mypal <- hcl.colors(nf) ; palpch <- 15:(nf + 15)
mycol <- mypal[myfac] ; mypch <- palpch[myfac]
plot(dfp$inter, coef(lmer3.2)$site[, 1], asp = 1, pch = mypch, bg = mycol, col = mycol,
     main = "Comparaison des ordonnées à l'origine", las = 1,
     xlab = "Modèle local UTC [h]", ylab = "Modèle mixte UTC [h]")
abline(c(0,1), col = "red2")
legend("bottomright", inset = 0.02, legend = levels(myfac), pch = palpch, pt.bg = mypal, col = mypal, title =
abline(c(-0.5, 1), lty = 2)
abline(c(1.5, 1), lty = 2)
```

Comparaison des ordonnées à l'origine



Je n'ai pas d'explication pour ce curieux résultat. Je ne pense pas que ce soit un artéfact dû à un problème de convergence : j'ai essayé diverses méthodes d'optimisation, mélangé les lignes de la table `dfq` et toujours convergé vers le même résultat. L'appel à `isSingular(lmer3.2)` renvoie d'ailleurs bien `FALSE`. Pour les 9 sites sur le méridien d'origine et les 18 sites à l'ouest d'icelui, les ordonnées à l'origine sont quasiment les mêmes et se posent sur la première bissectrice en rouge sur le graphe ci-dessus. Viennent ensuite les $3 \times 9 = 27$ sites immédiatement à l'est du méridien d'origine (entre $+7.5^\circ$ et $+22.5^\circ$ inclus) pour

lesquels la valeur de l'ordonnée à l'origine est inférieure de 0.5 h dans le modèle mixte par rapport au modèle local. Pour les 9 sites les plus orientaux (+30°) l'ordonnée à l'origine est au contraire supérieure de 1.5 h dans le modèle mixte par rapport au modèle local. Du point de vue de la tendance centrale, tout ceci finit par se compenser, et il n'y a pas de différence, aux arrondis numériques près, pour la moyenne des ordonnées à l'origine entre les deux modèles :

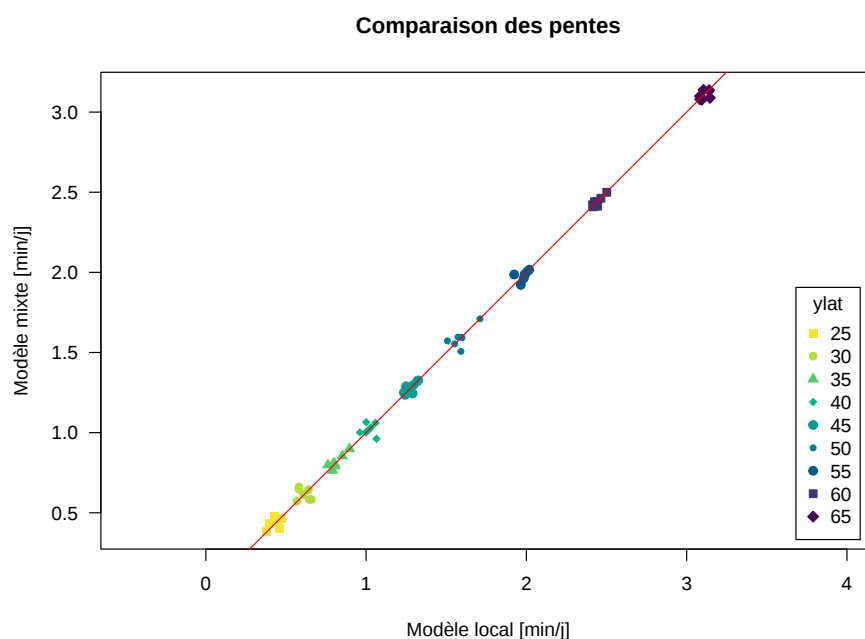
```
all.equal(mean(dfp$inter), mean(coef(lmer3.2)$site[ , 1]))
[1] TRUE
```

Il y aura donc forcément une différence au niveau de la dispersion des valeurs des ordonnées à l'origine entre les deux modèles. On constate que la dispersion est plus faible pour le modèle mixte que pour le modèle local. C'est l'effet de contraction, ou de régularisation, des modèles mixtes que nous avons évoqué dans la section 3.2 page 25.

```
sd(dfp$inter)
[1] 1.486494
sd(coef(lmer3.2)$site[ , 1])
[1] 1.482833
all.equal(sd(dfp$inter), sd(coef(lmer3.2)$site[ , 1]))
[1] "Mean relative difference: 0.002462829"
```

Au niveau des pentes on constate que les valeurs sont très voisines entre les deux modèles, même s'il y a des différences bien visibles :

```
myfac <- as.factor(dfp$ylat) ; nf <- length(levels(myfac))
mypal <- rev(hcl.colors(nf)) ; palpch <- 15:(nf + 15)
mycol <- mypal[myfac] ; mypch <- palpch[myfac]
plot(dfp$slp, 60*coef(lmer3.2)$site[ , 2], asp = 1, pch = mypch, bg = mycol, col = mycol,
     main = "Comparaison des pentes", las = 1,
     xlab = "Modèle local [min/j]", ylab = "Modèle mixte [min/j]")
abline(c(0,1), col = "red2")
legend("bottomright", inset = 0.02, legend = levels(myfac), pch = palpch, pt.bg = mypal, col = mypal, title =
```



COMME pour les ordonnées à l'origine, on constate que les moyennes des pentes sont identiques entre les deux modèles et que la dispersion est légèrement plus faible dans le cas du modèle mixte :

```
all.equal(mean(dfp$slp), 60*mean(coef(lmer3.2)$site[ , 2]))
[1] TRUE
sd(dfp$slp)
[1] 0.850892
sd(60*coef(lmer3.2)$site[ , 2])
[1] 0.848117
all.equal(sd(dfp$slp), sd(60*coef(lmer3.2)$site[ , 2]))
[1] "Mean relative difference: 0.003261309"
```

Références

- [1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and P. Czakı, editors, *Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akademiai Kiadó.
- [2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19 :716–723, 1974.
- [3] A. Basu, I.R. Harris, N.L. Hjort, and M.C. Jones. Robust and efficient estimation by minimizing a density power divergence. *Biometrika*, 85 :549–559, 1998.
- [4] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1) :1–48, 2015.
- [5] K.P. Burnham and D.R. Anderson. *Model Selection and Inference : A Practical Information-Theoretic Approach*. Springer-Verlag, New York, USA, 1998.
- [6] G. Claeskens and N.L. Hjort. The focused information criterion. *Journal of the American Statistical Association*, 98 :900–916, 2003.
- [7] G. Grolemond and H. Wickham. Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3) :1–25, 2011.
- [8] R.J. Hijmans. *terra : Spatial Data Analysis*, 2024. R package version 1.7-71.
- [9] C.M. Hurvich and C.-L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76 :297–307, 1989.
- [10] D. Lüdtcke, M.S. Ben-Shachar, I. Patil, P. Waggoner, and D. Makowski. performance : An R package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, 6(60) :3139, 2021.
- [11] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion - determining the number of hidden units for artificial neural network models. *IEEE Transactions on Neural Networks*, 5 :865–872, 1994.
- [12] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6 :461–464, 1978.
- [13] D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A. van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Ser. B*, 64 :583–639, 2002.
- [14] K. Takeuchi. Distribution of informational statistics and a criterion of model fitting (in japanese). *Suri-Kagaku (Mathematical Sciences)*, 153 :12–18, 1976.
- [15] B. Thieurmél and A. Elmarhraoui. *suncalc : Compute Sun Position, Sunlight Phases, Moon Position and Lunar Phase*, 2022. R package version 0.5.1.